Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft’s delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.

- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact dochelp@microsoft.com.
# Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/20/2012</td>
<td>0.1</td>
<td>New</td>
<td>Released new document.</td>
</tr>
<tr>
<td>4/11/2012</td>
<td>0.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>0.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/12/2012</td>
<td>0.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>1.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>2.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/30/2013</td>
<td>2.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>2.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>2.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>2.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>2.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>2.4</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/26/2016</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/15/2016</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/12/2017</td>
<td>3.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>6/20/2017</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
</tbody>
</table>
# Table of Contents

1 Introduction .................................................................................................................. 15

1.1 Glossary ....................................................................................................................... 15

1.2 References .................................................................................................................... 27

1.2.1 Normative References .......................................................................................... 27

1.2.2 Informative References ......................................................................................... 28

1.3 Overview ...................................................................................................................... 28

1.3.1 File Operations ....................................................................................................... 28

1.3.2 User and Group Operations .................................................................................. 28

1.4 Relationship to Other Protocols ............................................................................... 29

1.5 Prerequisites/Preconditions ...................................................................................... 29

1.6 Applicability Statement ............................................................................................. 29

1.7 Versioning and Capability Negotiation ........................................................................ 30

1.8 Vendor-Extensible Fields ........................................................................................... 30

1.9 Standards Assignments .............................................................................................. 30

2 Messages ....................................................................................................................... 31

2.1 Transport ..................................................................................................................... 31

2.2 Common Data Types ................................................................................................ 31

2.2.1 Simple Data Types and Enumerations ................................................................ 31

2.2.1.1 Simple Data Types .......................................................................................... 31

2.2.1.1.1 Content Type Identifier .............................................................................. 31

2.2.1.1.2 Document Identifier ................................................................................... 31

2.2.1.1.3 Event Receiver Identifier .......................................................................... 31

2.2.1.1.4 Feature Identifier ....................................................................................... 31

2.2.1.1.5 List Identifier .............................................................................................. 31

2.2.1.1.6 List Item Identifier ...................................................................................... 31

2.2.1.1.7 Role Identifier ............................................................................................. 31

2.2.1.1.8 Scope Identifier .......................................................................................... 32

2.2.1.1.9 Site Collection Identifier ........................................................................... 32

2.2.1.1.10 Site Group Identifier ................................................................................ 32

2.2.1.1.11 Site Identifier ........................................................................................... 32

2.2.1.1.12 SystemID ................................................................................................... 32

2.2.1.1.13 User Identifier .......................................................................................... 32

2.2.1.1.14 View Identifier .......................................................................................... 32

2.2.1.1.15 Web Part Identifier ................................................................................... 32

2.2.1.1.16 Workflow Identifier .................................................................................. 32

2.2.1.1.17 tPermMask ............................................................................................... 33

2.2.1.2 Enumerations ..................................................................................................... 33

2.2.1.2.1 Attachments Flag ...................................................................................... 33

2.2.1.2.2 Audit Item Type ......................................................................................... 33

2.2.1.2.3 Calendar Type ............................................................................................. 33

2.2.1.2.4 Collation Order Enumeration ...................................................................... 34

2.2.1.2.5 Event Host Type ......................................................................................... 36

2.2.1.2.6 Event Receiver Type ................................................................................... 36

2.2.1.2.7 Excluded Folder Type ................................................................................ 38

2.2.1.2.8 LinkDynamic Type ..................................................................................... 38

2.2.1.2.9 LinkSecurity Type ....................................................................................... 39

2.2.1.2.10 LinkType Types ......................................................................................... 39

2.2.1.2.11 List Base Type .......................................................................................... 40

2.2.1.2.12 List Server Template ............................................................................... 40

2.2.1.2.13 Moderation Status .................................................................................... 42

2.2.1.2.14 Page Type .................................................................................................. 43

2.2.1.2.15 Redirect Type ............................................................................................ 43

2.2.1.2.16 Role Definition Type .................................................................................. 43

2.2.1.2.17 Virus Status ............................................................................................... 44
2.2.1.2.18 App Principal Permissions Enumeration ................................................. 44
2.2.1.3 Time Zone Identifier ......................................................................................... 45
2.2.2 Bit Fields and Flag Structures .............................................................................. 48
  2.2.2.1 Audit Flags ........................................................................................................ 48
  2.2.2.2 Configuration Object Status ............................................................................ 49
  2.2.2.3 Doc Flags ......................................................................................................... 49
  2.2.2.4 Document Store Type ..................................................................................... 50
  2.2.2.5 List Flags ......................................................................................................... 51
  2.2.2.6 Publishing Level Type ..................................................................................... 54
  2.2.2.7 Put Flags Type ................................................................................................. 54
  2.2.2.8 Rename Flags ................................................................................................. 55
  2.2.2.9 Site Collection Flags ....................................................................................... 56
  2.2.2.10 Site Collection Upgrade Flags ...................................................................... 57
  2.2.2.11 Site Property Flags ....................................................................................... 57
  2.2.2.12 UserInfo tp_flags ......................................................................................... 58
  2.2.2.13 View Flags .................................................................................................... 58
  2.2.2.14 Workdays Flag .............................................................................................. 60
  2.2.2.15 WSS Rights Mask ........................................................................................ 60
2.2.3 Binary Structures .................................................................................................. 63
  2.2.3.1 Calendar View Options Type ........................................................................... 63
  2.2.3.2 External Group Token ..................................................................................... 64
  2.2.3.3 Token Group Offset and Attributes ................................................................. 65
  2.2.3.4 Token Groups .................................................................................................. 65
  2.2.3.5 WSS ACE ......................................................................................................... 66
  2.2.3.6 WSS ACL Format ............................................................................................ 66
  2.2.3.7 WSS External Group Map Cache Format ....................................................... 67
  2.2.3.8 WSS Compressed Structures ......................................................................... 67
  2.2.3.9 WSS External Group Record .......................................................................... 68
  2.2.3.10 WSS User Token ........................................................................................... 68
2.2.4 Result Sets ............................................................................................................. 69
  2.2.4.1 ACL and Permission Result Set ....................................................................... 69
  2.2.4.2 Custom Actions From Scope Result Set ......................................................... 69
  2.2.4.3 Domain Group Cache BEDS Update Result Set ............................................. 70
  2.2.4.4 Domain Group Cache Versions Result Set ...................................................... 70
  2.2.4.5 Domain Group Cache WFE Update Result Set .............................................. 71
  2.2.4.6 Document Content Metadata Result Set ....................................................... 71
  2.2.4.7 Document Content Stream Result Set ............................................................ 73
  2.2.4.8 Document Metadata Result Set ....................................................................... 73
  2.2.4.9 Document Version Metadata Result Set ........................................................ 77
  2.2.4.10 Empty Result Set ........................................................................................... 81
  2.2.4.11 Event Receivers Result Set .......................................................................... 81
  2.2.4.12 Individual URL Security Result Set ............................................................... 83
  2.2.4.13 Link Information Result Set .......................................................................... 85
  2.2.4.14 List Metadata Result Set ................................................................................ 85
  2.2.4.15 List Web Parts Result Set .............................................................................. 90
  2.2.4.16 NULL Individual URL Security Result Set .................................................... 91
  2.2.4.17 NULL Unique Permissions Result Set ............................................................ 91
  2.2.4.18 Object ID Result Set ...................................................................................... 92
  2.2.4.19 Principal User Information Result Set ............................................................ 92
  2.2.4.20 Server Time Result Set .................................................................................. 93
  2.2.4.21 Single Doc Link Information Result Set ......................................................... 93
  2.2.4.22 Site Audit Mask Result Set ............................................................................ 94
  2.2.4.23 Site Feature List Result Set ............................................................................ 94
  2.2.4.24 Site Metadata Result Set ................................................................................. 95
  2.2.4.25 Site MetaInfo Result Set ................................................................................ 100
  2.2.4.26 Unique Permissions Result Set ..................................................................... 100
  2.2.4.27 URL Result Set .............................................................................................. 101
  2.2.4.28 List Related Fields Result Set ....................................................................... 101

[MS-WSSFO3] - v20170620
Windows SharePoint Services (WSS): File Operations Database Communications Version 3 Protocol
Copyright © 2017 Microsoft Corporation
Release: June 20, 2017
2.2.4.29  App Principal Information Result Set .................................. 101
2.2.4.30  App Principal Permissions Result Set ................................... 102
2.2.5     SQL Structures .................................................................. 102
 2.2.5.1   Configuration Object ....................................................... 102
    2.2.5.1.1  Class ...................................................................... 102
    2.2.5.1.2  Id .......................................................................... 103
    2.2.5.1.3  Name .................................................................... 103
    2.2.5.1.4  Parent .................................................................... 103
    2.2.5.1.5  Status .................................................................... 104
    2.2.5.1.6  Version .................................................................. 104
    2.2.5.1.7  Properties .................................................................. 104
      2.2.5.1.7.1  Alternate URL Collection ....................................... 104
      2.2.5.1.7.2  Content Database ................................................ 105
      2.2.5.1.7.3  Web Application .................................................... 105
 2.2.5.2   Dependencies .................................................................. 106
2.2.6     Tables and Views ................................................................. 107
 2.2.6.1   AllDocs Table .................................................................. 107
 2.2.6.2   AllUserData Table ............................................................. 113
 2.2.6.3   Docs View ..................................................................... 174
 2.2.6.4   GroupMembership Table ................................................... 174
 2.2.6.5   Sec_SiteGroupsView ........................................................ 175
 2.2.6.6   AllSites Table ................................................................. 178
 2.2.6.7   Sites View ..................................................................... 182
 2.2.6.8   UserData View ............................................................... 182
 2.2.6.9   UserDataVersioned View .................................................. 182
 2.2.6.10  UserInfo Table ................................................................. 182
 2.2.6.11  WebMembers ................................................................. 184
 2.2.6.12  Versions ....................................................................... 184
2.2.7     XML Structures ................................................................ 185
 2.2.7.1   Namespaces .................................................................. 185
 2.2.7.2   Simple Types .................................................................. 185
    2.2.7.2.1  FALSE_CaseInsensitive_Else_Anything .............................. 185
    2.2.7.2.2  FieldInternalType ....................................................... 185
    2.2.7.2.3  FieldAggregationAttribute ........................................... 187
    2.2.7.2.4  FieldRefType ............................................................... 188
    2.2.7.2.5  FieldRichTextMode ..................................................... 189
    2.2.7.2.6  IMEMode ................................................................... 189
    2.2.7.2.7  IntPositive .................................................................. 189
    2.2.7.2.8  JoinType ................................................................... 190
    2.2.7.2.9  TextDirection .............................................................. 190
    2.2.7.2.10  TRUE_IF_Present ......................................................... 190
    2.2.7.2.11  TRUEFALSE ................................................................. 191
    2.2.7.2.12  TRUE_CaseSensitive_Else_Anything .............................. 191
    2.2.7.2.13  UniqueIdentifierWithBracesUppercase .......................... 191
    2.2.7.2.14  UniqueIdentifierWithOrWithoutBraces .......................... 191
    2.2.7.2.15  RelationshipDeleteBehaviorAttribute ............................ 191
 2.2.7.3    Complex Types ............................................................... 192
    2.2.7.3.1  CHOICEDEFINITION .................................................. 192
    2.2.7.3.1.1  Schema .................................................................. 192
    2.2.7.3.1.2  Attributes ............................................................... 192
    2.2.7.3.1.3  Child Elements .......................................................... 192
    2.2.7.3.2  CHOICEDEFINITIONS ................................................ 192
    2.2.7.3.2.1  Schema .................................................................. 192
    2.2.7.3.2.2  Attributes ............................................................... 192
    2.2.7.3.2.3  Child Elements .......................................................... 192
    2.2.7.3.3  FieldDefinition ............................................................ 193
    2.2.7.3.3.1  Schema .................................................................. 193
    2.2.7.3.3.2  Attributes ............................................................... 196
3 Protocol Details ........................................................................................................212

3.1 Back-End Database Server Details .................................................................212

3.1.1 Abstract Data Model ......................................................................................212

3.1.2 Timers ............................................................................................................212
3.1.3 Initialization .......................................................... 213
3.1.4 Higher-Layer Triggered Events .................................. 213
3.1.5 Message Processing Events and Sequencing Rules ............ 213
   3.1.5.1 fn_GetFullUrl ............................................. 222
   3.1.5.2 proc_AddBuildDependency ................................ 223
   3.1.5.3 proc_AddDocument ....................................... 223
   3.1.5.3.1 Site List for Normalization Result Set ............... 227
   3.1.5.3.2 Checkout Information Result Set ....................... 227
   3.1.5.4 proc_AddListItem ....................................... 228
   3.1.5.5 proc_ChangeLevelForDoc ................................ 233
   3.1.5.6 proc_CheckRbsInstalled ................................ 235
   3.1.5.7 proc_CheckoutDocument .................................. 235
   3.1.5.7.1 Link Info Single Doc Result Set ..................... 237
   3.1.5.7.2 Document Metadata Result Set ....................... 237
   3.1.5.7.3 Event Receivers Result Set ........................... 238
   3.1.5.7.4 Audit Mask Result Set ................................ 238
   3.1.5.8 proc_ClearLinks ......................................... 238
   3.1.5.9 proc_CreateDir .......................................... 238
   3.1.5.10 proc_DeleteAllDocumentVersions ....................... 241
   3.1.5.11 proc_DeleteDocBuildDependencySet .................... 242
   3.1.5.12 proc_DeleteDocumentVersion ............................ 243
   3.1.5.13 proc_DeleteUrl ......................................... 244
   3.1.5.13.1 Deleted Documents Result Set ...................... 246
   3.1.5.13.2 Deleted Aliased Lists Result Set ................... 246
   3.1.5.13.3 Empty Deleted Aliased Lists Result Set ............. 247
   3.1.5.14 proc_DisableRbs ....................................... 247
   3.1.5.15 proc_DirtyDependents ................................ 248
   3.1.5.16 proc_EnableRbs ........................................ 248
   3.1.5.17 proc_NumLists ......................................... 249
   3.1.5.17.1 List Information Result Set ....................... 250
   3.1.5.17.2 Recycle Bin Information Result Set ................ 253
   3.1.5.17.3 NULL Result Set ..................................... 254
   3.1.5.18 proc_ReadStream ....................................... 254
   3.1.5.18.1 Document Stream Chunk Result Set ................ 255
   3.1.5.19 procFetchRequestHttpGet ................................ 255
   3.1.5.19.1 HTTP Document Metadata Result Set ................ 258
   3.1.5.19.2 Domain Group Cache Versions Result Set ........... 261
   3.1.5.19.3 Domain Group Cache BDES Update Result Set ........ 261
   3.1.5.19.4 Domain Group Cache WFE Update Result Set .......... 262
   3.1.5.19.5 User Information Result Set ........................ 262
   3.1.5.19.6 Welcome Page Redirect Information Result Set .... 263
   3.1.5.19.7 Non-Welcome Page Redirect Information Result Set . 263
   3.1.5.19.8 Document Information (Get) Metadata Result Set ... 263
   3.1.5.19.9 Document Information (Get) Stream Result Set ...... 264
   3.1.5.19.10 Site Collection Audit Mask Result Set .......... 264
   3.1.5.19.11 List Audit Mask Result Set ........................ 264
   3.1.5.19.12 Document Build Dependency Set Result Set ........ 264
   3.1.5.19.13 Document Build Dependency Metadata Result Set ... 265
   3.1.5.19.14 Site Metadata Result Set .......................... 266
   3.1.5.19.15 Event Receivers Result Set ........................ 267
   3.1.5.19.16 Web Event Receiver Result Set .................... 267
   3.1.5.19.17 Site Features List Result Set ........................ 267
   3.1.5.19.18 WebParts Metadata, Personalized Result Set ...... 267
   3.1.5.19.19 Web Parts Metadata, Nonpersonalized Result Set ... 269
   3.1.5.19.20 List Metadata Result Set ........................... 271
   3.1.5.19.21 List Event Receivers Result Set ................... 271
   3.1.5.19.22 List Security Information Result Set .............. 272
   3.1.5.19.23 SiteCollection Custom Action Result Set ........... 272
3.1.5.19.24 Site Custom Actions Result Set .............................................272
3.1.5.19.25 List Custom Actions Result Set .............................................272
3.1.5.19.26 List Web Parts Result Set ...................................................273
3.1.5.19.27 Content Type Order Result Set ..........................................273
3.1.5.19.28 Current Folder Scope Result Set .........................................273
3.1.5.19.29 Navigation Context Security Information Result Set ...............274
3.1.5.19.30 NULL Navigation Context Security Information Result Set ........274
3.1.5.19.31 Empty Navigation Context Security Information Result Set .......274
3.1.5.19.32 App Principal Information Result Set .....................................274
3.1.5.19.33 App Principal Permissions Result Set .....................................274
3.1.5.20 proc_FetchDocForRead ..........................................................275
3.1.5.20.1 Subsite List Result Set .........................................................277
3.1.5.20.2 Link Info Single Doc Result Set .............................................277
3.1.5.20.3 Document Metadata Result Set .............................................277
3.1.5.20.4 Document Version Metadata Result Set ................................277
3.1.5.20.5 NULL Result Set ...................................................................277
3.1.5.20.6 Event Receivers Result Set ....................................................277
3.1.5.20.7 List Metadata Result Set .......................................................278
3.1.5.20.8 Empty List Result Set ...........................................................278
3.1.5.20.9 Event Receivers Result Set (1) ..............................................278
3.1.5.20.10 Document Information and Content (Read) Metadata Result Set 278
3.1.5.20.11 Document Information and Content (Read) Stream Result Set ...279
3.1.5.20.12 Document Version Information and Content Result Set ...........279
3.1.5.20.13 Document Version Information and Content Stream Result Set ...279
3.1.5.20.14 Attachment State Result Set .................................................279
3.1.5.20.15 Audit Mask Result Set .........................................................279
3.1.5.21 proc_FetchDocForUpdate .........................................................280
3.1.5.21.1 Subsite List Result Set ..........................................................281
3.1.5.21.2 ACL and Permission Result Set .............................................281
3.1.5.21.3 Document Metadata Result Set .............................................282
3.1.5.21.4 Document Version Metadata Result Set ................................282
3.1.5.21.5 NULL Result Set ...................................................................282
3.1.5.21.6 Event Receivers Result Set ....................................................282
3.1.5.21.7 Link Info Single Doc Fixup Result Set ....................................282
3.1.5.21.8 Web Part Info Result Set .......................................................283
3.1.5.21.9 Zone ID Result Set ...............................................................284
3.1.5.21.10 File Format Metadata Result Set ..........................................284
3.1.5.21.11 List Metadata Result Set .....................................................284
3.1.5.21.12 Empty List Result Set ..........................................................284
3.1.5.21.13 Event Receivers Result Set (1) ..............................................285
3.1.5.21.14 Document Information and Content (Update) Metadata Result Set 285
3.1.5.21.15 Document Information and Content (Update) Stream Result Set ...285
3.1.5.21.16 Document Version Information and Content (Update) Metadata Result Set 285
3.1.5.21.17 Document Version Information and Content (Update) Stream Result Set 285
3.1.5.21.18 Attachment State Result Set ................................................286
3.1.5.22 proc_FetchWelcomeNames ......................................................286
3.1.5.22.1 Welcome Pages Result Set ....................................................286
3.1.5.23 proc_GenerateNextId ............................................................287
3.1.5.24 proc_GetAllRolesForUser ......................................................287
3.1.5.24.1 User Roles Result Set ...........................................................288
3.1.5.25 proc_GetAuditMask ...............................................................288
3.1.5.25.1 Audit Mask Result Set ..........................................................288
3.1.5.26 proc_GetAuditMaskOutput ....................................................289
3.1.5.27 proc_GetContainingList ......................................................289
3.1.5.27.1 List Metadata Result Set .....................................................290
3.1.5.27.2 Empty Result Set ...............................................................290
3.1.5.27.3 Event Receivers Result Set ...................................................290
3.1.5.28 proc_GetContainingListCore .................................................. 291
3.1.5.28.1 List Metadata Result Set .................................................. 292
3.1.5.28.2 Empty Result Set ............................................................. 292
3.1.5.29 proc_GetDocsMetalInfo ......................................................... 292
3.1.5.29.1 Individual URL Security Result Set ...................................... 294
3.1.5.29.2 NULL Individual Url Security Result Set ............................... 294
3.1.5.29.3 Server Time Result Set ..................................................... 294
3.1.5.29.4 Subsite List Result Set ..................................................... 294
3.1.5.29.5 Link Info Result Set ....................................................... 295
3.1.5.29.6 Multiple Document Metadata Result Set .............................. 295
3.1.5.30 proc_GetLinkInfoSingleDoc ................................................... 299
3.1.5.30.1 Link Info Single Doc Result Set ......................................... 299
3.1.5.31 proc_GetListCheckedOutFiles ............................................... 299
3.1.5.31.1 Checked Out Files Result Set ............................................. 300
3.1.5.32 proc_GetListFields ............................................................. 300
3.1.5.32.1 Fields Information Result Set ............................................ 301
3.1.5.33 proc_GetListMetaDataAndEventReceivers .................................. 301
3.1.5.33.1 List Metadata Result Set .................................................. 302
3.1.5.33.2 Unique Permissions Result Set .......................................... 302
3.1.5.33.3 NULL Unique Permissions Result Set .................................... 302
3.1.5.33.4 List Event Receivers Result Set ........................................ 303
3.1.5.33.5 List Web Parts Result Set ................................................ 303
3.1.5.33.6 List Related Fields Result Set .......................................... 303
3.1.5.34 proc_getObject ................................................................. 303
3.1.5.34.1 Object Result Set ........................................................... 303
3.1.5.35 proc_getObjectsByBaseClass ............................................... 304
3.1.5.35.1 Object ID Result Set ....................................................... 304
3.1.5.36 proc_getObjectsByClass ....................................................... 305
3.1.5.36.1 Object ID Result Set ....................................................... 305
3.1.5.37 proc_getSiteFlags ............................................................... 305
3.1.5.37.1 Site Collection Flags Result Set ........................................ 306
3.1.5.38 proc_getSiteMap ................................................................. 306
3.1.5.38.1 Site Map Result Set ........................................................ 306
3.1.5.39 proc_getSiteMapById .......................................................... 307
3.1.5.39.1 Site Map By Id Result Set ................................................ 307
3.1.5.40 proc_GetTpWebMetaDataAndListMetaData .................................. 308
3.1.5.40.1 Web Url Result Set ......................................................... 311
3.1.5.40.2 Domain Group Cache Versions Result Set .............................. 312
3.1.5.40.3 Domain Group Cache BEDS Update Result Set ........................ 312
3.1.5.40.4 Domain Group Cache WFE Update Result Set .......................... 312
3.1.5.40.5 Site Metadata Result Set ................................................ 312
3.1.5.40.6 Site Collection Event Receivers Result Set .......................... 313
3.1.5.40.7 Site Event Receivers Result Set ........................................ 313
3.1.5.40.8 Site MetaInfo Result Set ................................................. 313
3.1.5.40.9 Site Feature List Result Set ............................................. 313
3.1.5.40.10 Site Unique Permissions Result Set .................................... 314
3.1.5.40.11 Site NULL Unique Permissions Result Set ............................. 314
3.1.5.40.12 Empty Result Set .......................................................... 314
3.1.5.40.13 Redirect Url Result Set .................................................. 314
3.1.5.40.14 No Welcome Redirect Url Result Set ................................... 315
3.1.5.40.15 List Identifier Result Set ............................................... 315
3.1.5.40.16 List Metadata Result Set ................................................ 316
3.1.5.40.17 List Related Fields Result Set ........................................ 316
3.1.5.40.18 List Unique Permissions Result Set .................................... 316
3.1.5.40.19 List NULL Unique Permissions Result Set ............................ 316
3.1.5.40.20 Event Receivers Result Set ............................................. 317
3.1.5.40.21 List Web Parts Result Set ............................................... 317
3.1.5.40.22 List Metadata Result Set (1) .......................................... 317
3.1.5.40.23 Unique Permissions Result Set ..........................................................317
3.1.5.40.24 NULL Unique Permissions Result Set ..................................................318
3.1.5.40.25 Event Receivers Result Set (1) ..........................................................318
3.1.5.40.26 List Web Parts Result Set (1) .............................................................319
3.1.5.40.27 Document Metadata Result Set ..........................................................319
3.1.5.40.28 NULL Result Set .................................................................................319
3.1.5.40.29 App Principal Information Result Set ..................................................319
3.1.5.40.30 App Principal Permissions Result Set ..................................................320
3.1.5.41 proc_GetUniqueScopesInList ..................................................................320
3.1.5.41.1 Unique Permissions Result Set ..............................................................320
3.1.5.41.2 NULL Unique Permissions Result Set .....................................................320
3.1.5.42 proc_GetVersion ......................................................................................321
3.1.5.43 proc_GetWebMetaInfo .............................................................................321
3.1.5.43.1 Site MetaInfo Result Set .......................................................................322
3.1.5.43.2 Domain Group Cache Versions Result Set .............................................323
3.1.5.43.3 Domain Group Cache BEDS Update Result Set .....................................323
3.1.5.43.4 Domain Group Cache WFE Update Result Set .......................................323
3.1.5.43.5 Site Metadata Result Set .......................................................................324
3.1.5.43.6 Event Receivers Result Set ...................................................................324
3.1.5.44 proc_GetWebMetaInfoByUrl .................................................................324
3.1.5.44.1 Site URL Result Set ..............................................................................325
3.1.5.44.2 Site Metainfo Result Set ........................................................................325
3.1.5.44.3 Domain Group Cache Versions Result Set ............................................325
3.1.5.44.4 Domain Group Cache BEDS Update Result Set ....................................325
3.1.5.44.5 Domain Group Cache WFE Update Result Set ........................................326
3.1.5.44.6 Site Metadata Result Set .......................................................................326
3.1.5.44.7 Event Receivers Result Set ...................................................................326
3.1.5.45 proc_ListDocumentVersions ..................................................................326
3.1.5.45.1 Individual URL Security Result Set ......................................................327
3.1.5.45.2 NULL Individual URL Security Result Set .............................................327
3.1.5.45.3 Document Versions Result Set ..............................................................327
3.1.5.46 proc_ListRbsStores ..................................................................................328
3.1.5.46.1 List RBS Stores Result Set ....................................................................329
3.1.5.47 proc_ListUrls .........................................................................................329
3.1.5.47.1 Individual URL Security Result Set ......................................................330
3.1.5.47.2 NULL Individual URL Security Result Set .............................................330
3.1.5.47.3 Server Time Result Set .........................................................................330
3.1.5.47.4 Subsite List Result Set ..........................................................................331
3.1.5.47.5 Document Metadata Result Set ............................................................331
3.1.5.47.6 Link Info Result Set ..............................................................................334
3.1.5.47.7 Contained Document Metadata Result Set ...........................................335
3.1.5.48 proc_RenameUrl .....................................................................................339
3.1.5.48.1 Rename Result Set ..............................................................................341
3.1.5.48.2 Backward Link Result Set ....................................................................342
3.1.5.49 proc_SecAddPrincipalToRole .................................................................342
3.1.5.49.1 Site Audit Mask Result Set ....................................................................343
3.1.5.50 proc_SecAddRoleDef ..............................................................................343
3.1.5.50.1 Site Audit Mask Result Set ....................................................................344
3.1.5.51 proc_SecAddUser ....................................................................................344
3.1.5.52 proc_SecAddUserToSiteGroup ................................................................347
3.1.5.53 proc_SecAddWebMembership .................................................................348
3.1.5.54 proc_SecChangeToInheritedList ..............................................................348
3.1.5.54.1 Site Audit Mask Result Set ....................................................................349
3.1.5.55 proc_SecChangeToInheritedWeb ..............................................................349
3.1.5.55.1 Inherited Site Result Set .......................................................................350
3.1.5.55.2 Site Audit Mask Result Set ....................................................................350
3.1.5.56 proc_SecChangeToUniqueScope ..............................................................350
3.1.5.56.1 Site Audit Mask Result Set ....................................................................352
3.1.5.57 proc_SecCheckDeletedAccounts
3.1.5.57.1 Login Result Set
3.1.5.58 proc_SecCloneRoleDefinitions
3.1.5.58.1 Site Audit Mask Result Set
3.1.5.59 proc_SecCreateSiteGroup
3.1.5.60 proc_SecDecCurrentUsersCount
3.1.5.61 proc_SecGetAccountStatus
3.1.5.61.1 Account Status Result Set
3.1.5.62 proc_SecGetAclFromScope
3.1.5.62.1 ACL and Permission Result Set
3.1.5.63 proc_SecGetAllAclsForSite
3.1.5.63.1 Access Control List Result Set
3.1.5.64 proc_SecGetAllGroupsAndMembershipInfo
3.1.5.64.1 Groups Result Set
3.1.5.64.2 Group Membership Result Set
3.1.5.65 proc_SecGetApplicationPrincipalAndUserToken
3.1.5.65.1 Application Principal Result Set
3.1.5.65.2 External Token Result Set
3.1.5.66 proc_SecGetCompleteWebRoleMemberList
3.1.5.66.1 Role Member Result Set
3.1.5.67 proc_SecGetCurrentUsersCount
3.1.5.67.1 User Count Result Set
3.1.5.68 proc_SecGetDomainGroupMapData
3.1.5.68.1 Domain Group Cache Versions Result Set
3.1.5.68.2 Domain Group Cache BEDS Update Result Set
3.1.5.68.3 Domain Group Cache WFE Update Result Set
3.1.5.69 proc_SecGetGroupById
3.1.5.70 proc_SecGetGroupOwner
3.1.5.71 proc_SecGetGroupSecurityScopes
3.1.5.71.1 Security Scopes Result Set
3.1.5.72 proc_SecGetIndividualUrlSecurityCheckEventReceivers
3.1.5.72.1 Individual URL Security Result Set
3.1.5.72.2 NULL Individual URL Security Result Set
3.1.5.72.3 List Metadata Result Set
3.1.5.72.4 Event Receivers Result Set
3.1.5.72.5 List Scopes Result Set
3.1.5.73 proc_SecGetItemsWithUniquePermissions
3.1.5.73.1 Items with Unique Permissions Result Set
3.1.5.74 proc_SecGetPrincipalByEmail
3.1.5.74.1 Principal User Information Result Set
3.1.5.75 proc_SecGetPrincipalByGuid
3.1.5.75.1 User Information Result Set
3.1.5.76 proc_SecGetPrincipalByGuidEx
3.1.5.76.1 Extended Principal User Information Result Set
3.1.5.77 proc_SecGetPrincipalByLogin
3.1.5.77.1 User Information Result Set
3.1.5.78 proc_SecGetPrincipalByLogin20
3.1.5.78.1 User Information Result Set
3.1.5.79 proc_SecGetPrincipal(DisplayInformation20
3.1.5.79.1 Site Group Principal Display Information Result Set
3.1.5.79.2 Security Principal Display Information Result Set
3.1.5.80 proc_SecGetRoleAssignments
3.1.5.80.1 WSSACE Result Set
3.1.5.81 proc_SecGetRoleBindingsForAllPrincipals
3.1.5.81.1 Role Assignment Result Set
3.1.5.82 proc_SecGetRoleDefs
3.1.5.82.1 Role Definition Result Set
3.1.5.83 proc_SecGetSecurityInfo
3.1.5.83.1 Security Information Result Set ........................................ 380
3.1.5.84 proc_SecGetSiteAdmins .................................................. 380
3.1.5.84.1 Site Administrators Result Set ..................................... 381
3.1.5.85 proc_SecGetSiteGroupById ............................................. 381
3.1.5.85.1 Site Group Result Set ................................................ 382
3.1.5.86 proc_SecGetSiteGroupByTitle ......................................... 382
3.1.5.86.1 Site Group Information Result Set .................................. 382
3.1.5.87 proc_SecGetSiteGroupByTitle20 ....................................... 382
3.1.5.87.1 Site Group Information Result Set .................................. 383
3.1.5.88 proc_SecGetUserAccountDirectoryPath ............................... 383
3.1.5.88.1 User Account Directory Path Result Set .......................... 383
3.1.5.89 proc_SecGetUserPermissionOnGroup .................................. 384
3.1.5.90 proc_SecGetWebsAndListWithUniquePermissions ..................... 385
3.1.5.90.1 Site Information Result Set ......................................... 385
3.1.5.90.2 Sub Site Information Result Set .................................... 385
3.1.5.90.3 Site List Information Result Set .................................... 386
3.1.5.90.4 Sub Site List Information Result Set ................................ 386
3.1.5.91 proc_SecListAllSiteMembers .......................................... 387
3.1.5.91.1 User Information Result Set ........................................ 387
3.1.5.92 proc_SecListAllWebMembers ........................................... 388
3.1.5.92.1 Site Membership Result Set ........................................ 388
3.1.5.93 proc_SecListGroupInRole ............................................... 389
3.1.5.93.1 Site Group Information Result Set .................................. 389
3.1.5.94 proc_SecListScopeGroups .............................................. 389
3.1.5.94.1 Site Groups Result Set .............................................. 390
3.1.5.95 proc_SecListScopeUsers ............................................... 390
3.1.5.95.1 User Information Result Set ........................................ 390
3.1.5.96 proc_SecListSiteGroupMembership ................................... 390
3.1.5.96.1 User Information Result Set ........................................ 391
3.1.5.97 proc_SecListSiteGroups ............................................... 391
3.1.5.97.1 Site Group Information Result Set .................................. 392
3.1.5.98 proc_SecListSiteGroupsContainingUser ............................... 392
3.1.5.98.1 Site Group Information Result Set .................................. 392
3.1.5.99 proc_SecListSiteGroupsWhichUserOwns ................................ 392
3.1.5.99.1 Site Group Information Result Set .................................. 393
3.1.5.100 proc_SecListSiteUsersInRole ........................................ 393
3.1.5.100.1 User Role Membership Result Set ................................ 393
3.1.5.101 proc_SecMigrateUser ................................................. 394
3.1.5.102 proc_SecReCalculateWebFGP ........................................ 395
3.1.5.103 proc_SecRefreshToken ............................................... 395
3.1.5.104 proc_SecRemoveGroup ............................................... 396
3.1.5.104.1 DLAlias Result Set ................................................ 397
3.1.5.105 proc_SecMarkGroupForWebDelete .................................... 397
3.1.5.105.1 DLAlias Result Set ................................................ 398
3.1.5.106 proc_SecRemovePrincipalFromScope ................................ 399
3.1.5.106.1 Site Audit Mask Result Set ....................................... 400
3.1.5.107 proc_SecRemoveRoleDef .............................................. 400
3.1.5.107.1 Site Audit Mask Result Set ....................................... 400
3.1.5.108 proc_SecRemoveUserFromScopeByLogin .............................. 400
3.1.5.108.1 Site Audit Mask Result Set ....................................... 401
3.1.5.109 proc_SecRemoveUserFromSite ........................................ 402
3.1.5.110 proc_SecRemoveUserFromSiteGroup .................................. 402
3.1.5.111 proc_SecRemoveUserFromSiteGroupByLogin ........................ 403
3.1.5.112 proc_SecResetItemPerm ............................................... 405
3.1.5.112.1 Site Audit Mask Result Set ....................................... 405
3.1.5.113 proc_SecResetWebToDefaultRoleDefinition ......................... 405
3.1.5.113.1 Site Audit Mask Result Set ....................................... 407
3.1.5.114 proc_SecResolvePrincipal ............................................ 407
3.1.5.114.1 Principal Information Result Set ..................................................... 408
3.1.5.115 proc_SecSetSiteGroupProperties ...................................................... 409
3.1.5.116 proc_SecSetUserAccountDirectoryPath ............................................. 411
3.1.5.117 proc_SecSetWebRequestAccess .......................................................... 411
3.1.5.118 proc_SecUpdateAnonymousPermMask ............................................... 412
3.1.5.119 proc_SecUpdateDomainGroupMapData ............................................ 412
3.1.5.120 proc_SecUpdateRoleDef ................................................................. 413
3.1.5.120.1 Site Audit Mask Result Set ......................................................... 414
3.1.5.121 proc_SecUpdateUser ................................................................. 414
3.1.5.122 proc_SecUpdateUserActiveStatus .................................................... 415
3.1.5.123 proc_TakeOverCheckOut ............................................................... 416
3.1.5.124 proc_UncheckoutDocument ............................................................. 416
3.1.5.124.1 Link Info Result Set ........................................................................ 417
3.1.5.124.2 Document Metadata Result Set ..................................................... 418
3.1.5.124.3 NULL Result Set ............................................................................. 418
3.1.5.124.4 Event Receivers Result Set ........................................................... 418
3.1.5.125 proc_UpdateDocBuildDependencySet ............................................... 418
3.1.5.126 proc_UpdateDocument ....................................................................... 419
3.1.5.126.1 Site List For Normalization Result Set ......................................... 424
3.1.5.126.2 Site Audit Mask Result Set ............................................................ 424
3.1.5.126.3 Lock Information Result Set .......................................................... 424
3.1.5.127 proc_UpdateListItem ........................................................................ 425
3.1.5.127.1 Item Update Result Set .................................................................. 429
3.1.5.128 proc_UpdateListSettings ................................................................. 430
3.1.5.129 proc_UpdateUserInfoInTableFromRowUpdater .................................. 434
3.1.5.130 proc_UserToWebUrl ......................................................................... 436
3.1.5.130.1 Web URL Result Set ....................................................................... 436
3.1.5.131 proc_WriteChunkToAllDocStreams .................................................. 436
3.1.5.132 proc_WriteStreamToRBS .................................................................. 441
3.1.5.133 proc_WriteStreams ........................................................................... 442
3.1.5.134 proc_WriteStreamToSQL .................................................................. 442
3.1.5.135 proc_SetStreamsToDoc ...................................................................... 444
3.1.5.136 proc_SetStreamsToDocNoTVP .......................................................... 445
3.1.5.137 TVF_Docs_Url_Level ........................................................................ 447
3.1.5.138 TVF_UserData_ListItemLevelRow .................................................... 447
3.1.5.139 TVF_UserData_PId_DId_Level_Row ............................................... 448
3.1.5.140 proc_HasCurrentPublishVersion ....................................................... 449
3.1.5.141 proc_GetSiteDenyPermMask ............................................................ 449
3.1.5.141.1 Deny Permissions Mask Result Set ................................................ 449
3.1.5.142 proc_GetNewListItemId .................................................................. 450
3.1.5.143 proc_UpdateDiskUsed ..................................................................... 450
3.1.6 Timer Events ............................................................................................... 451
3.1.7 Other Local Events ...................................................................................... 451
3.2 Web Front End Client Details ......................................................................... 451
3.2.1 Abstract Data Model ................................................................................ 451
3.2.2 Timers ........................................................................................................ 452
3.2.3 Initialization .............................................................................................. 452
3.2.4 Higher-Layer Triggered Events ............................................................... 452
3.2.5 Message Processing Events and Sequencing Rules .................................. 452
3.2.6 Timer Events ............................................................................................... 452
3.2.7 Other Local Events ...................................................................................... 452

4 Protocol Examples ............................................................................................ 453
4.1 File: Open File OM ....................................................................................... 453
4.1.1 Determine a User's Permission Level to a Document ............................... 454
4.2 Group Add User to Site Group OM ............................................................. 455
4.3 Group Update Site Group Properties OM .................................................... 457
4.4 Security: Add User to Document Library via Object Model ........................ 459
4.5 Security: Break Web Inheritance OM ................................................................. 461
4.6 Site Collection Lookup ....................................................................................... 462
  4.6.1 Retrieving the Farm Id .................................................................................. 462
  4.6.2 Retrieving the Alternate URL Collection Ids ............................................... 463
  4.6.3 Retrieving the Alternate URL Collections .................................................... 463
  4.6.4 Alternate URL Matching ............................................................................. 463
  4.6.5 Retrieving the Web Service Ids .................................................................... 463
  4.6.6 Retrieving the Web Application Ids ............................................................... 464
  4.6.7 Retrieving the Web Applications .................................................................. 464
  4.6.8 Web Application Lookup ............................................................................ 464
  4.6.9 Prefix Matching ........................................................................................... 465
    4.6.9.1 Explicit Prefixes ...................................................................................... 465
    4.6.9.2 Wildcard Prefixes ................................................................................... 465
  4.6.10 Site Collection Id Lookup ............................................................................ 466
  4.6.11 Building Content Database Connection String ............................................ 466
    4.6.11.1 Name .................................................................................................... 466
4.7 User Update User Properties OM ....................................................................... 467
4.8 Version Negotiation ............................................................................................ 469
4.9 Folder: Move Folder OM ..................................................................................... 470

5 Security .................................................................................................................. 472
  5.1 Security Considerations for Implementers ....................................................... 472
  5.2 Index of Security Parameters ............................................................................ 472

6 Appendix A: Product Behavior .............................................................................. 475

7 Change Tracking .................................................................................................... 476

8 Index ......................................................................................................................... 477
1 Introduction

The File Operations Database Communications Version 3 Protocol specifies the communication sequences used by front-end Web servers to perform data query and update commands on back-end database servers as part of file, user, and group administration operations.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

- **access control list (ACL)**: A list of access control entries (ACEs) that collectively describe the security rules for authorizing access to some resource; for example, an object or set of objects.

- **Active Directory**: A general-purpose network directory service. Active Directory also refers to the Windows implementation of a directory service. Active Directory stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in Active Directory. Active Directory is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS). [MS-ADTS] describes both forms. For more information, see [MS-AUTHSOD] section 1.1.1.5.2, Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Kerberos, and DNS.

- **Active Directory account creation mode**: A type of account creation mode that retrieves and uses user accounts in a specific Active Directory Domain Services (AD DS) organizational unit.

- **alert**: An Internet message that is sent to subscribers automatically to notify them when user-defined criteria are met. Alerts are generated automatically when items such as documents, webpages, list items, sites, or other resources on a server are changed.

- **anonymous user**: A user who presents no credentials when identifying himself or herself. The process for determining an anonymous user can differ based on the authentication protocol, and the documentation for the relevant authentication protocol should be consulted.

- **app**: See web application.

- **app host header name**: A unique name assigned to an app instance.

- **app principal**: Designates an authenticated entity that is not a user.

- **app site domain identifier**: A unique specifier containing six hexadecimal number values that is used to determine a particular host header for a subsite.

- **app web domain identifier**: A unique specifier containing eight hexadecimal number values that is used to determine a particular host header for a subsite.

- **ASP.NET**: A web server technology for dynamically rendering HTML pages using a combination of HTML, Javascript, CSS, and server-side logic. For more information, see [ASPNET].

- **assembly name**: The name of a collection of one or more files that is versioned and deployed as a unit. See also assembly.

- **attachment**: An external file that is included with an Internet message or associated with an item in a SharePoint list.

- **audit entry**: Information that is recorded about an operation on an object that is stored on a server.
Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [RFC5234].

author: The user who created a list item.

back-end database server: A server that hosts data, configuration settings, and stored procedures that are associated with one or more applications.

backward link: A hyperlink between a referenced document and a referencing party. For example, if Document A contains a hyperlink to Document B, then Document B has a backward link to Document A.

binary large object (BLOB): A discrete packet of data that is stored in a database and is treated as a sequence of uninterpreted bytes.

blog: A website that contains a series of posts about a subject and is arranged in reverse chronological order. Also referred to as web log.

Boolean: An operation or expression that can be evaluated only as either true or false.

bot: A structured HTML comment that is processed by a front-end web server when the containing document is opened by or saved to the server. Also referred to as web bot.

build dependency set: A serialized .NET Framework object that represents a set of file dependencies.

calculated column: A column in a table that contains a formula that is copied automatically to each record in the column.

cascading style sheet (CSS): An extension to HTML that enables authors and users of HTML documents to attach style sheets to those documents, as described in [CSS-LEVEL1] and [CSS-LEVEL2]. A style sheet includes typographical information about the appearance of a page, including the font for text on the page.

Central Administration site: A SharePoint site that an administrator can use to manage all of the sites and servers in a server farm that is running SharePoint Products and Technologies.

change log: A log of changes, such as add and delete, that are made to objects that are stored on a back-end database server. Applications can use this information to identify changes that occurred on those objects.

character set: The range of characters used to represent textual data within a MIME body part, as described in [RFC2046].

check in: The process of placing a file or project into a source repository. This releases the lock for editing and enables other users to view the updated file or check out the file. See also check out.

check out: The process of retrieving a writable copy of a file or project from a source repository. This locks the file for editing to prevent other users from overwriting or editing it inadvertently. See also check in.

checked out: A publishing level that indicates that a document has been created and locked for exclusive editing by a user in a version control system.

class identifier (CLSID): A GUID that identifies a software component; for instance, a DCOM object class or a COM class.
**code page**: An ordered set of characters of a specific script in which a numerical index (code-point value) is associated with each character. Code pages are a means of providing support for **character sets** and keyboard layouts used in different countries. Devices such as the display and keyboard can be configured to use a specific code page and to switch from one code page (such as the United States) to another (such as Portugal) at the user’s request.

**Collaborative Application Markup Language (CAML)**: An XML-based language that is used to describe various elements, such as queries and views, in sites that are based on SharePoint Products and Technologies.

**collation order**: A rule for establishing a sequence for textual information.

**content database**: A database that is stored on a **back-end database server** and contains stored procedures, site collections, and the contents of those site collections.

**content type**: A named and uniquely identifiable collection of settings and fields that store metadata for individual items in a SharePoint list. One or more content types can be associated with a list, which restricts the contents to items of those types.

**content type identifier**: A unique identifier that is assigned to a **content type**.

**Collaborative Universal Time (UTC)**: A high-precision atomic time standard that approximately tracks Universal Time (UT). It is the basis for legal, civil time all over the Earth. Time zones around the world are expressed as positive and negative offsets from UTC. In this role, it is also referred to as Zulu time (Z) and Greenwich Mean Time (GMT). In these specifications, all references to UTC refer to the time at UTC-0 (or GMT).

**crawler**: A process that browses and indexes content from a content source.

**current user**: The user who is authenticated during processing operations on a **front-end web server** or a **back-end database server**.

**current version**: The latest version of a document that is available to a user, based on the permissions of the user and the publishing level of the document.

**custom action**: An extension to the user interface, such as a button on a toolbar or a link on a site settings page.

**customized**: A document, column, or content type whose content is stored in a content database instead of a front-end file system. Also referred to as unghosted.

**datasheet**: A worksheet window that contains the source data for a Microsoft Graph chart object.

**datetime**: A data type that represents the date and time when a document can be normalized and indexed as a numeric value by a search application. The range and degree of granularity varies according to search application and implementation.

**default view**: The layout and organization of a document or list that appears automatically when users open that document or display that list.

**directory name**: A segment of a **store-relative URL** that refers to a directory. A directory name is everything that appears before the last slash in a store-relative form URL.

**dirty**: The condition of an entity, such as a component or a file, that indicates that the entity or properties of the entity were changed after the entity was last saved.

**discussion board**: A list in which users can read, post, and reply to messages from other users who are members of the same discussion board.

**display name**: A text string that is used to identify a principal or other object in the user interface. Also referred to as title.
displayed version: Document version information that is formatted for display in the user interface. The displayed version uses the format MajorVersion.MinorVersion, where MajorVersion is the published version number and MinorVersion is the draft version number, separated by a decimal point. See also major version and minor version.

distribution list: A collection of users, computers, contacts, or other groups that is used only for email distribution, and addressed as a single recipient.

document: An object in a content database such as a file, folder, list (1), or site (2). Each object is identified by a URI.

document flag: A 4-byte unsigned integer bit mask that provides metadata about the document.

document identifier: A GUID that identifies a document.

document library: A type of list that is a container for documents and folders.

document stream: A byte stream that is associated with a document, such as the content of a file. Some documents do not have document streams.

document template: A file that serves as the basis for new documents.

document version: A copy of a list item that has a version number. A document version can be either a historical version or a current version.

domain group: A container for security and distribution groups. A domain group can also contain other domain groups.

draft: A version of a document or list item that does not have a publishing level of "Published" or "Checked Out".

draft: A version of a document or list that does not have a publishing level of "Published" or "Checked Out".

editor: The user who last modified an item or document in a SharePoint list.

email address: A string that identifies a user and enables the user to receive Internet messages.

email enabled list: A SharePoint list that is configured to accept incoming email messages.

event: (1) Any significant occurrence in a system or an application that requires users to be notified or an entry to be added to a log.

(2) An action or occurrence to which an application might respond. Examples include state changes, data transfers, key presses, and mouse movements.

event host: A site collection, site (2), list (1), list item, workflow, feature, or content type that hosts an event receiver.

event receiver: A structured modular component that enables built-in or user-defined managed code classes to act upon objects, such as list items, lists (1), or content types, when specific triggering actions occur.

event sink: A structured, modular component that enables built-in or user-defined classes to act on documents in document libraries when specific triggering actions occur. Event sinks are a deprecated, implementation-specific capability of Windows SharePoint Services 2.0. In Windows SharePoint Services 3.0 and Microsoft SharePoint Foundation 2010, they are replaced by the capabilities of event receivers.

expire: A process in which an object, such as an external data connection, becomes invalid because its allotted time period has ended.

external security provider: An external object that manages permissions on a site.
**feature**: A package of SharePoint elements that can be activated or deactivated for a specific feature scope.

**feature identifier**: A **GUID** that identifies a **feature**.

**field**: A container for metadata within a SharePoint list and associated list items.

**field definition**: The definition of a field in the **Collaborative Application Markup Language (CAML)**.

**field identifier**: A **GUID** that is used to identify a field.

**file**: A single, discrete unit of content.

**file extension**: The sequence of characters in a **file's** name between the end of the **file's** name and the last "." character. Vendors of applications choose such sequences for the applications to uniquely identify **files** that were created by those applications. This allows file management software to determine which application are to be used to open a **file**.

**folder**: A file system construct. File systems organize a volume's data by providing a hierarchy of objects, which are referred to as folders or directories, that contain files and can also contain other folders.

**form**: A document with a set of controls into which users can enter information. Controls on a form can be bound to elements in the data source of the form, such as fields and groups. See also **bind**.

**forward link**: A hyperlink between a referenced document and a referencing party. For example, if Document A contains a hyperlink to Document B, Document A has a forward link to Document B.

**front-end web server**: A server that hosts webpages, performs processing tasks, and accepts requests from protocol clients and sends them to the appropriate back-end server for further processing.

**full URL**: A string of characters in a standardized format that identifies a document or resource on the World Wide Web.

**fully qualified class name**: A class name that includes namespace information. Use of a fully qualified class name ensures that the class name is treated as unique.

**globally unique identifier (GUID)**: A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

**group**: (1) An element that can contain fields and other groups in the data source for an InfoPath form. Controls that contain other controls, such as repeating tables and sections, are bound to groups.

(2) A named collection of users who share similar access permissions or roles.

**hash**: A fixed-size result that is obtained by applying a one-way mathematical function, which is sometimes referred to as a hash algorithm, to an arbitrary amount of data. If the input data changes, the hash also changes. The hash can be used in many operations, including authentication and digital signing.

**home page**: On the World Wide Web, a document that serves as a starting point for a set of webpages and other files in a website.
**host header**: An Internet host and port number that identifies a network resource.

**host name**: The name of a physical server, as described in [RFC952].

**HTTP GET**: An HTTP method for retrieving a resource, as described in [RFC2616].

**HTTP HEAD**: An HTTP method for retrieving header information for a resource, as described in [RFC2616].

**Hypertext Markup Language (HTML)**: An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [HTML].

**Hypertext Transfer Protocol (HTTP)**: An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

**Hypertext Transfer Protocol Secure (HTTPS)**: An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

**Input Method Editor (IME)**: An application that is used to enter characters in written Asian languages by using a standard 101-key keyboard. An IME consists of both an engine that converts keystrokes into phonetic and ideographic characters and a dictionary of commonly used ideographic words.

**internal version number**: A number that increases monotonically and is used to identify conflicts when saving an item.

**item**: A unit of content that can be indexed and searched by a search application.

**item identifier**: An integer that uniquely identifies an item in a SharePoint list.

**language code identifier (LCID)**: A 32-bit number that identifies the user interface human language dialect or variation that is supported by an application or a client computer.

**leaf name**: The segment of a URL that follows the last slash. If the resource is a directory, the leaf name can be an empty string.

**level**: A relative position in a hierarchy of data. A level is frequently used when describing how to navigate a hierarchy in an Online Analytical Processing (OLAP) database or a PivotTable report.

**list**: (1) A container within a SharePoint site that stores list items. A list has a customizable schema that is composed of one or more fields.

(2) An organization of a region of cells into a tabular structure in a workbook.

**list identifier**: A GUID that is used to identify a list (1) in a site collection.

**list item**: An individual entry within a SharePoint list. Each list item has a schema that maps to fields in the list that contains the item, depending on the content type of the item.

**list item identifier**: See item identifier.

**list server template**: A value that identifies the template that is used for a SharePoint list.

**list template**: An XML-based definition of list settings, including fields and views, and optionally list items. List templates are stored in .stp files in the content database.

**list template identifier**: A GUID that is used to identify a list template for a SharePoint list.
**list view**: A named collection of settings for querying and displaying items in a SharePoint list. There are two types of views: Personal, which can be used only by the user who created the view; and Public, which can be used by all users who have permission to access to the site.

**locale**: A collection of rules and data that are specific to a language and a geographical area. A locale can include information about sorting rules, date and time formatting, numeric and monetary conventions, and character classification.

**locked**: The condition of a cell, worksheet, or other object that restricts edits or modifications to it by users.

**login name**: A string that is used to identify a user or entity to an operating system, directory service, or distributed system. For example, in Windows-integrated authentication, a login name uses the form "DOMAIN\username".

**major version**: An iteration of a software component, document, or list item that is ready for a larger group to see, or has changed significantly from the previous major version. For an item on a SharePoint site, the minor version is always "0" (zero) for a major version.

**manifest**: A file that stores metadata about an expansion pack, such as the name of the expansion pack, the files and resources that are included in the expansion pack, and the dependencies that it has on other files and components.

**master page**: An ASP.NET file that has a predefined layout that can include static text, HTML elements, and server controls.

**meeting instance**: A collection of data for a meeting that occurs only once or a single occurrence of a meeting that occurs multiple times. The data can be stored in a client application or on a website.

**Meeting Workspace site**: A SharePoint site that is based on a Meeting Workspace site template and has a template ID value of "2". A Meeting Workspace site is used for planning, posting, and working together on meeting materials.

**member**: A user in the Members group of a site.

**metadict**: A dictionary that has strongly typed values.

**minor version**: An iteration of a software component, document, or list item that is in progress or has changed only slightly from the previous version. For an item on a SharePoint site, the minor version number is never "0" (zero) and is incremented for each new version of an item, unless a major version is explicitly published. When minor versioning is disabled on a SharePoint site, only major version numbers are incremented, and the minor version is always "0" (zero).

**mobile device**: A small computing device that is easily portable and can be used in various environments.

**moderation status**: A content approval status that indicates whether a list item was approved by a moderator.

**navigation node**: An element in the navigational structure of a site. The element is a link or a series of links to a specific page in the site.

**navigation node element identifier**: An integer that identifies a navigation node. This value is unique for every navigation node in the navigational structure of a SharePoint site.

**navigation structure**: A hierarchical organization of links between related content on a site.

**NULL GUID**: A GUID of all zeros.

**owner**: A security principal who has the requisite permission to manage a security group.
page: A file that consists of HTML and can include references to graphics, scripts, or dynamic content such as Web Parts.

parent site: The site that is above the current site in the hierarchy of the site collection.

path segment: A portion of a URI, as described in [RFC3986]. See also path component.

permission: A rule that is associated with an object and that regulates which users can gain access to the object and in what manner. See also rights.

permission level: A set of permissions that can be granted to principals or SharePoint groups on an entity such as a site, list, folder, item, or document.

personal view: A view of a list that is created by a user for personal use. The view is unavailable to other users.

principal: (1) An authenticated entity that initiates a message or channel in a distributed system.

(2) An identifier of such an entity.

provisioned: A condition of an object that was created and deployed successfully.

published: A condition of portions of a workbook that are marked as being available to the user when that workbook is processed by a protocol server.

published version: The version of a list item that is approved and can be seen by all users. The user interface (UI) version number for a published version is incremented to the next positive major version number and the minor version is "0" (zero). See also major version and minor version.

publishing level: An integer that is assigned to a document to indicate the publishing status of that version of the document.

read-only mode: An attribute that indicates that an object cannot be changed or deleted. The object can only be accessed or displayed.

Really Simple Syndication (RSS): An XML-based syndication format for content, as described in [RSS2.0].

Recycle Bin: The location where deleted files are stored until they are either restored, if they were deleted erroneously, or destroyed permanently.

request identifier: A GUID that is used to identify a specific action or procedure that is sent to a protocol server or a protocol client.

result set: A list of records that results from running a stored procedure or query, or applying a filter. The structure and content of the data in a result set varies according to the implementation.

return code: A code that is used to report the outcome of a procedure or to influence subsequent events when a routine or process terminates (returns) and passes control of the system to another routine. For example, a return code can indicate whether an operation was successful.

rich text: Text that is formatted in the Rich Text Format, as described in [MSFT-RTF].

role: A symbolic name that defines a class of users for a set of components. A role defines which users can call interfaces on a component.

role assignment: An association between a principal or a site group and a role definition.

role definition: A named set of permissions for a SharePoint site. See also permission level.
**role identifier**: An integer that uniquely identifies a role definition within a SharePoint site.

**root folder**: The folder at the top of a hierarchy of folders in a list.

**sandboxed solution**: A custom solution that can be deployed to a site by a site collection administrator, without approval from the server farm administrator.

**scope identifier**: A GUID that uniquely identifies a scope within a site collection.

**securable object**: An object that can have unique security permissions associated with it.

**Secure Sockets Layer (SSL)**: A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL uses two keys to encrypt data—a public key known to everyone and a private or secret key known only to the recipient of the message. SSL supports server and, optionally, client authentication using X.509 certificates. For more information, see [X509]. The SSL protocol is precursor to Transport Layer Security (TLS). The TLS version 1.0 specification is based on SSL version 3.0 [SSL3].

**security group**: A named group of principals on a SharePoint site.

**security identifier (SID)**: An identifier for security principals that is used to identify an account or a group. Conceptually, the SID is composed of an account authority portion (typically a domain) and a smaller integer representing an identity relative to the account authority, termed the relative identifier (RID). The SID format is specified in [MS-DTYP] section 2.4.2; a string representation of SIDs is specified in [MS-DTYP] section 2.4.2 and [MS-AZOD] section 1.1.1.2.

**security policy**: In the form of a collection of security policy settings, the policy itself is an expression of administrative intent regarding how computers and resources on their network should be secured.

**security principal**: An identity that can be used to regulate access to resources. A security principal can be a user, a computer, or a group that represents a set of users.

**security provider**: A Component Object Model (COM) object that provides methods that return custom information about the security of a site.

**security role**: A defined set of access privileges. The security role that is assigned to a user determines the tasks that a user can perform and which parts of the user interface a user can view.

**security scope**: A tree structure of objects in which every object has the same security settings as the root.

**server-relative URL**: A relative URL that does not specify a scheme or host, and assumes a base URI of the root of the host, as described in [RFC3986].

**setup path**: The location where supporting files for a product or technology are installed.

**site**: (1) A group of related webpages that is hosted by a server on the World Wide Web or an intranet. Each website has its own entry points, metadata, administration settings, and workflows. Also referred to as web site.

   (2) A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

**site collection**: A set of websites that are in the same content database, have the same owner, and share administration settings. A site collection can be identified by a GUID or the URL of the top-level site for the site collection. Each site collection contains a top-level site, can contain one or more subsites, and can have a shared navigational structure.
site collection administrator: A user who has administrative permissions for a site collection.

site collection identifier: A GUID that identifies a site collection. In stored procedures, the identifier is typically "@SiteId" or "@WebSiteId". In databases, the identifier is typically "SiteId/tp_SiteId".

site definition: A family of site definition configurations. Each site definition specifies a name and contains a list of associated site definition configurations.

site identifier: A GUID that is used to identify a site in a site collection.

site subscription identifier: A GUID that is used to identify a site subscription.

site template: An XML-based definition of site settings, including formatting, lists, views, and elements such as text, graphics, page layout, and styles. Site templates are stored in .stp files in the content database.

site-relative URL: A URL that is relative to the site that contains a resource and does not begin with a leading slash (/).

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

stored procedure: A precompiled collection of SQL statements and, optionally, control-of-flow statements that are stored under a name and processed as a unit. They are stored in a SQL database and can be run with one call from an application. Stored procedures return an integer return code and can additionally return one or more result sets. Also referred to as sproc.

store-relative form: See store-relative URL.

store-relative URL: A URL that consists only of a path segment and does not include the leading and trailing slash.

stream: An element of a compound file, as described in [MS-CFB]. A stream contains a sequence of bytes that can be read from or written to by an application, and they can exist only in storages.

stream binary piece: A block of binary data containing the length (in bytes) of a data stream and the data stream itself.

stream identifier: A 64-bit integer that uniquely specifies a stream binary piece.

stream partition: A byte that identifies the container for a stream binary piece.

stream schema: A numeric selector that specifies the format of a file's stream binary pieces.

Structured Query Language (SQL): A database query and programming language that is widely used for accessing, querying, updating, and managing data in relational database systems.

subsite: A complete website that is stored in a named subdirectory of another website. The parent website can be the top-level site of a site collection or another subsite. Also referred to as subweb.

SystemID: A binary identifier that is used to uniquely identify a security principal. For Windows integrated authentication, it is a security identifier (SID). For an ASP.NET Forms Authentication provider, it is the binary representation that is derived from a combination of the provider name and the user login name.
TCP/IP: A set of networking protocols that is widely used on the Internet and provides communications across interconnected networks of computers with diverse hardware architectures and various operating systems. It includes standards for how computers communicate and conventions for connecting networks and routing traffic.

thicket: A means of storing a complex HTML document with its related files. It consists of a thicket main file and a hidden thicket folder that contains a thicket manifest and a set of thicket supporting files that, together, store the referenced content of the document.

thicket folder: A hidden folder that contains a thicket manifest and a set of thicket supporting files that, together, store the referenced content of a complex HTML document.

thicket main file: The core file of a complex HTML document. It references contained elements such as graphics, pictures, or other media that are stored as thicket supporting files in a thicket folder. The thicket main file is the target that is used by a protocol client to access the content of the document.

thicket supporting file: A file that contains a graphic element, a picture, or other media that is referenced by the thicket main file and is stored in the thicket folder.

Transact-Structured Query Language (T-SQL): A language that contains the commands that are used to manage instances of Microsoft SQL Server, create and manage all objects in an instance of SQL Server, and to insert, retrieve, modify, and delete all data in SQL Server tables. Transact-SQL is an extension of the language that is defined in the SQL standards that are published by the International Standards Organization (ISO) and the American National Standards Institute (ANSI).

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of messages in client and server applications communicating over open networks. TLS supports server and, optionally, client authentication by using X.509 certificates (as specified in [X509]). TLS is standardized in the IETF TLS working group.

uncustomized: A condition of a document whose content is stored in a location other than the content database. If a document is uncustomized, the front-end web server determines the location of the content by using the SetupPath value for the document. Also referred to as ghosted.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

upgrade evaluation site collection: A copy of the current site collection, used to evaluate the functionality of a site collection after it is upgraded.

user account directory path: A string representation of the Lightweight Directory Access Protocol (LDAP) distinguished name for an AD DS container. It defines a set of users, as described in [RFC4514].

user identifier: An integer that uniquely identifies a security principal as distinct from all other security principals and site groups within the same site collection.

user information list: A list that contains items, each of which represents a security principal in a site collection. Each site collection has only one such list and it resides in the top-level site of the site collection.

user interface (UI) version: A single 4-byte integer that stores the version number that appears as a document version number in the user interface. The lower 9 bits correspond to the minor
version number of the displayed version. The remaining 23 bits correspond to the major version number of the displayed version. See also displayed version.

user name: A unique name that identifies a specific user account. The user name of an account is unique among the other group names and user names within its own domain or workgroup.

UTF-8: A byte-oriented standard for encoding Unicode characters, defined in the Unicode standard. Unless specified otherwise, this term refers to the UTF-8 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

version: See displayed version, historical version, major version, and minor version.

view: See form view (Microsoft InfoPath), list view (SharePoint Products and Technologies), or View (Microsoft Business Connectivity Services).

virus scanner: Software that is used to search for and remove computer viruses, worms, and Trojan horses.

web application: A container in a configuration database that stores administrative settings and entry-point URLs for site collections.

web bot: See bot.

Web Part: A reusable component that contains or generates web-based content such as XML, HTML, and scripting code. It has a standard property schema and displays that content in a cohesive unit on a webpage. See also Web Parts Page.

Web Part zone: A structured HTML section of a Web Parts Page that contains zero or more Web Parts and can be configured to control the organization and format of those Web Parts.

Web Part zone identifier: A string that identifies a Web Part zone on a Web Parts Page.

website: A group of related webpages that is hosted by a server on the World Wide Web or an intranet. Each website has its own entry points, metadata, administration settings, and workflows. Also referred to as site.

Windows code page: A table that relates the character codes (code point values) that are used by an application to keys on a keyboard or to characters on a display. This provides support for character sets and keyboard layouts for different countries or regions. Also referred to as character set or charset.

Windows collation name: A string identifier that follows the format of the Transact-Structured Query Language (T-SQL) COLLATE clause.

workflow: A structured modular component that enables the automated movement of documents or items through a specific sequence of actions or tasks that are related to built-in or user-defined business processes.

workflow identifier: A GUID that is used to identify a workflow.

workflow instance: An instance of a workflow association that performs on a list item the process that is defined in a workflow template.

XML: The Extensible Markup Language, as described in [XML1.0].

XML document: A document object that is well formed, as described in [XML10/5], and might be valid. An XML document has a logical structure that is composed of declarations, elements, comments, character references, and processing instructions. It also has a physical structure that is composed of entities, starting with the root, or document, entity.
**XML namespace**: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

**XML Path Language (XPath)**: A language used to create expressions that can address parts of an XML document, manipulate strings, numbers, and Booleans, and can match a set of nodes in the document, as specified in [XPATH]. XPath models an XML document as a tree of nodes of different types, including element, attribute, and text. XPath expressions can identify the nodes in an XML document based on their type, name, and values, as well as the relationship of a node to other nodes in the document.

**XML schema definition (XSD)**: The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring XML schemas.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT**: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

### 1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

#### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

- [MS-DTYP] Microsoft Corporation, "Windows Data Types".
- [MS-TDS] Microsoft Corporation, "Tabular Data Stream Protocol".
1.3 Overview

This protocol specifies the communication between the front-end Web server and the back-end database server used to satisfy requests involving file access and administration of users and groups (2). This client-to-server protocol uses the Tabular Data Stream Protocol [MS-TDS] as its transport between the front-end Web server, acting as a client, and the back-end database server, acting as a server. The T-SQL language [TSQL-Ref] is used to define the queries and returned data which is transported over Tabular Data Stream.

End User Clients use remote file access protocols to communicate with the Windows SharePoint Services front-end Web servers, specifically using FrontPage Server Extensions Remote Protocol (as specified in [MS-FPSE]), Hypertext Transfer Protocol (HTTP) (as specified in [RFC2616]), and WebDAV (as specified in [RFC2518]).

Further information about the interoperation of the clients with the front-end Web server, and the front-end Web server with the back-end database server, can be found in the Windows SharePoint Services Technical Document [MS-WSSO].

1.3.1 File Operations

This protocol provides methods for retrieving and manipulating files' properties, along with support for retrieving and manipulating files' security information. When client requests for files or file information are sent to the front-end Web server, the front-end Web server sends a series of stored procedure calls to the back-end database server for the requested information. The stored procedures return data that in turn can be used for further calls to other stored procedures. The front-end Web server turns the values in the stored procedures' return codes and result sets into the data and metadata for the files requested by the client, and sends it back to the client using the same protocol used by the initial request.

1.3.2 User and Group Operations

This protocol provides methods for retrieving and manipulating information about individual users and groups (2), along with support for retrieving information from Active Directory about users. When the Object Model on the front-end Web server operates on requests to query or update users or groups (2), the front-end Web server confirms whether the data is already populated in the local objects that represent the specific user or groups (2). If it does not exist, the front-end Web server sends a series of stored procedure calls to the back-end database server for the requested information. The stored procedures return data, which in turn can be used for further calls to other
stored procedures. The front-end Web server turns the values in the stored procedures' return codes and result sets into objects that contain the data and metadata for the requested users or groups (2), and uses the objects according to implementation-specific procedures.

1.4 Relationship to Other Protocols

This protocol relies on [MS-TDS] as its transport protocol to invoke stored procedures to inspect and manipulate document properties via result sets and return codes. Database queries or calls to stored procedures, and the returned result sets, are written in the [TSQL-Ref] language.

The relationship between Tabular Data Stream Protocol (TDS), SNMP Multiplexing (SMUX), Transport Layer Security (TLS) and Secure Sockets Layer (SSL), and Transmission Control Protocol/Internet Protocol (TCP/IP) is illustrated in the following figure.

![Figure 1: Protocol relationship](image)

Requests to a front-end Web server via FrontPage Server Extensions (as specified in [MS-FPSE]) and WebDAV (as specified in [RFC2518]) rely on this protocol, via the front-end Web server, to retrieve and manipulate file and security information persistently stored on the back-end database server and to service requests for files and their properties from their clients.

1.5 Prerequisites/Preconditions

Unless otherwise specified, the stored procedures and any related tables are present in the content database that is being queried on the back-end database server. The tables in the content database have to contain valid data in a consistent state to be queried successfully by the stored procedures.

For operations defined in this document, any file access, addition, or modification has to be to a valid location, such as a site (2), list, document library, folder, or document, as defined by the data within the tables and the front-end Web server, in order for the request to be successfully processed. The user making the request to the front-end Web server has to have adequate permission to access the content of the specified valid location in order for the request to be successfully processed.

1.6 Applicability Statement

This protocol is only applicable to front-end Web servers when communicating with the back-end database server for file, user, and group (2) administration operations.
1.7 Versioning and Capability Negotiation

The front-end Web server and back-end database server use this protocol to perform explicit version verifications. The front-end Web server calls stored procedure **proc_GetVersion** to retrieve version information from the back-end database server, which it uses to decide whether it needs to connect with the back-end database server. The version information is stored in the Versions table (section 2.2.6.12).

1.8 Vendor-Extensible Fields

For complex types defined in this document, an implementation of protocol servers can optionally support reading, writing, and persisting additional arbitrary elements or attributes. Unless otherwise specified in the underlying schema definition, this capability MUST NOT be used by vendors or by protocol server implementations to provide extensions to the schema specified in this document.

The **Customization** element, which is specified in the **FieldDefinition** type (section 2.2.7.3.3), can be used by third parties to store additional data on a **field**.

1.9 Standards Assignments

None.
2 Messages

2.1 Transport

[MS-TDS] is the transport protocol used to call the stored procedures, query SQL Views, or SQL Tables, and return Result Codes and result sets.

2.2 Common Data Types

The following are common data types used in conjunction with this protocol. The low-level data type and size are specified using commonly-known data type descriptions. It is possible that the variable is stored in multiple T-SQL data types, depending on the actual implementation of each Stored Procedure, Result Set, or Database Table.

2.2.1 Simple Data Types and Enumerations

2.2.1.1 Simple Data Types

2.2.1.1.1 Content Type Identifier

A content type identifier is a numeric string value of arbitrary but limited length, which uniquely identifies a content type, stored on the back-end database server (BEDS) as a T-SQL varbinary(512).

2.2.1.1.2 Document Identifier

A document identifier is a GUID, as specified in [MS-DTYP] section 2.3.2.3, used to uniquely identify a document within a site collection. Specialized varieties of document identifiers include site identifiers and list identifiers.

2.2.1.1.3 Event Receiver Identifier

An event receiver identifier is a GUID used to uniquely identify an event receiver within a site collection.

2.2.1.1.4 Feature Identifier

A feature identifier is a GUID used to uniquely identify a feature within a site collection.

2.2.1.1.5 List Identifier

A list identifier is a variety of document identifier. The list identifier is a GUID used to uniquely identify a list within a site collection.

2.2.1.1.6 List Item Identifier

A list item identifier is a 4-byte integer value used to uniquely identify a list item within any list in a particular site collection.

2.2.1.1.7 Role Identifier

A Role Identifier is a 4-byte integer value used to uniquely identify a Role Definition within a site collection.
<table>
<thead>
<tr>
<th>Role identifier value</th>
<th>Role definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1073741825</td>
<td>Guest</td>
</tr>
<tr>
<td>1073741826</td>
<td>Reader</td>
</tr>
<tr>
<td>1073741827</td>
<td>Contributor</td>
</tr>
<tr>
<td>1073741828</td>
<td>Web Designer</td>
</tr>
<tr>
<td>1073741829</td>
<td>Administrator</td>
</tr>
</tbody>
</table>

### 2.2.1.1.8 Scope Identifier

A **scope identifier** is a GUID used to uniquely identify a scope within a site collection.

### 2.2.1.1.9 Site Collection Identifier

A site collection identifier is a GUID used to uniquely identify a site collection within a content database.

### 2.2.1.1.10 Site Group Identifier

A site group identifier is a 4-byte integer value used to uniquely identify a site group within a site collection. Site group identifiers are assigned from the same numbering space as User Identifiers and cannot overlap. Values of -1 and 0 are reserved to indicate invalid or unknown user or site group identifiers.

### 2.2.1.1.11 Site Identifier

A site identifier is a variety of **document identifier**. The site identifier is a GUID used to uniquely identify a site within a site collection.

### 2.2.1.1.12 SystemID

A **SystemID** is a binary value of arbitrary but limited length that uniquely identifies a **principal**, stored on the BEDS as a T-SQL varbinary(512).

### 2.2.1.1.13 User Identifier

A **user identifier** is a 4-byte integer value used to uniquely identify a principal within a site collection. User identifiers are assigned from the same numbering space as site group identifiers and cannot overlap. Certain user identifiers are predefined such as the system account (1073741823), the site collection owner, and the secondary site collection contact (2). Values of -1 and 0 are reserved to indicate invalid or unknown user or site group identifiers.

### 2.2.1.1.14 View Identifier

A view identifier is a 4-byte integer value used to identify a **view** within a list or document library. A view identifier is unique only within a particular list or document library.

### 2.2.1.1.15 Web Part Identifier

A Web Part identifier is a GUID used to uniquely identify a **Web Part** within a site collection.

### 2.2.1.1.16 Workflow Identifier
A **workflow identifier** is a GUID used to uniquely identify a **workflow** within a site collection.

### 2.2.1.1.17 tPermMask

A **tPermMask** is an 8-byte integer value used to specify the rights that can be assigned to a user or site group.

### 2.2.1.2 Enumerations

#### 2.2.1.2.1 Attachments Flag

The **Attachments Flag** is a 1-byte integer flag that specifies whether an item appears to be an attachment or a folder related to attachments based on this document’s **Uniform Resource Locator (URL)**. The following are valid values for **Attachments Flag**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The URL does not appear to be an attachment.</td>
</tr>
<tr>
<td>1</td>
<td>The URL is an attachment file. The directory name of the document has the string “Attachments” as its next-to-last <strong>path segment</strong>, and a 32-bit, base-10, signed integer as the last path segment that is referring to the item ID to which this file is attached and where the permissions will be checked (for example, &quot;Announcements/Attachments/17/file1.txt&quot;).</td>
</tr>
<tr>
<td>2</td>
<td>The URL is a list item attachment folder (for example, &quot;Announcements/Attachments/17&quot;).</td>
</tr>
<tr>
<td>3</td>
<td>The URL is the list Attachments folder itself. The last path segment of the URL is the string &quot;Attachments&quot; (for example, &quot;Announcements/Attachments&quot;).</td>
</tr>
</tbody>
</table>

#### 2.2.1.2.2 Audit Item Type

**Audit Item Type** is a 2-byte integer flag that specifies the type of the object in an audit specification. The following are the only valid values for **Audit Item Type**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A page or a file.</td>
</tr>
<tr>
<td>3</td>
<td>A list item.</td>
</tr>
<tr>
<td>4</td>
<td>A list.</td>
</tr>
<tr>
<td>5</td>
<td>A folder.</td>
</tr>
<tr>
<td>6</td>
<td>A site (2).</td>
</tr>
<tr>
<td>7</td>
<td>A site collection.</td>
</tr>
</tbody>
</table>

#### 2.2.1.2.3 Calendar Type

**Calendar Type** is a 2-byte integer value that specifies the type of calendar to use in a particular context. The only valid values of the **Calendar Type** are specified as follows.
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gregorian (localized)</td>
</tr>
<tr>
<td>3</td>
<td>Japanese Emperor Era</td>
</tr>
<tr>
<td>4</td>
<td>Taiwan Calendar</td>
</tr>
<tr>
<td>5</td>
<td>Korean Tangun Era</td>
</tr>
<tr>
<td>6</td>
<td>Hijri (Arabic Lunar)</td>
</tr>
<tr>
<td>7</td>
<td>Thai</td>
</tr>
<tr>
<td>8</td>
<td>Hebrew (Lunar)</td>
</tr>
<tr>
<td>9</td>
<td>Gregorian (Middle East French)</td>
</tr>
<tr>
<td>10</td>
<td>Gregorian (Arabic)</td>
</tr>
<tr>
<td>11</td>
<td>Gregorian (Transliterated English)</td>
</tr>
<tr>
<td>12</td>
<td>Gregorian (Transliterated French)</td>
</tr>
<tr>
<td>14</td>
<td>Korean and Japanese Lunar</td>
</tr>
<tr>
<td>15</td>
<td>Chinese Lunar</td>
</tr>
<tr>
<td>16</td>
<td>Saka Era</td>
</tr>
<tr>
<td>23</td>
<td>Umm al-Qura</td>
</tr>
</tbody>
</table>

2.2.1.2.4 Collation Order Enumeration

Collation Order Enumeration is a 2-byte integer value indicating collation order for textual information. The Collation Order values are mapped to Windows Collation Designator values, as defined in [Iseminger]. The only valid values of Collation Order Enumeration are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Albanian</td>
</tr>
<tr>
<td>1</td>
<td>Arabic</td>
</tr>
<tr>
<td>2</td>
<td>Chinese_PRC</td>
</tr>
<tr>
<td>3</td>
<td>Chinese_PRC_Stroke</td>
</tr>
<tr>
<td>4</td>
<td>Chinese_Taiwan_Bopomofo</td>
</tr>
<tr>
<td>5</td>
<td>Chinese_Taiwan_Stroke</td>
</tr>
<tr>
<td>6</td>
<td>Croatian</td>
</tr>
<tr>
<td>7</td>
<td>Cyrillic_General</td>
</tr>
<tr>
<td>8</td>
<td>Czech</td>
</tr>
<tr>
<td>9</td>
<td>Danish_Norwegian</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>10</td>
<td>Estonian</td>
</tr>
<tr>
<td>11</td>
<td>Finnish_Swedish</td>
</tr>
<tr>
<td>12</td>
<td>French</td>
</tr>
<tr>
<td>13</td>
<td>Georgian_Modern_Sort</td>
</tr>
<tr>
<td>14</td>
<td>German_PhoneBook</td>
</tr>
<tr>
<td>15</td>
<td>Greek</td>
</tr>
<tr>
<td>16</td>
<td>Hebrew</td>
</tr>
<tr>
<td>17</td>
<td>Hindi</td>
</tr>
<tr>
<td>18</td>
<td>Hungarian</td>
</tr>
<tr>
<td>19</td>
<td>Hungarian_Technical</td>
</tr>
<tr>
<td>20</td>
<td>Icelandic</td>
</tr>
<tr>
<td>21</td>
<td>Japanese</td>
</tr>
<tr>
<td>22</td>
<td>Japanese_Unicode</td>
</tr>
<tr>
<td>23</td>
<td>Korean_Wansung</td>
</tr>
<tr>
<td>24</td>
<td>Korean_Wansung_Unicode</td>
</tr>
<tr>
<td>25</td>
<td>Latin1_General</td>
</tr>
<tr>
<td>26</td>
<td>Latvian</td>
</tr>
<tr>
<td>27</td>
<td>Lithuanian</td>
</tr>
<tr>
<td>28</td>
<td>Lithuanian_Classic</td>
</tr>
<tr>
<td>29</td>
<td>Traditional_Spanish</td>
</tr>
<tr>
<td>30</td>
<td>Modern_Spanish</td>
</tr>
<tr>
<td>31</td>
<td>Polish</td>
</tr>
<tr>
<td>32</td>
<td>Romanian</td>
</tr>
<tr>
<td>33</td>
<td>Slovak</td>
</tr>
<tr>
<td>34</td>
<td>Slovenian</td>
</tr>
<tr>
<td>35</td>
<td>Thai</td>
</tr>
<tr>
<td>36</td>
<td>Turkish</td>
</tr>
<tr>
<td>37</td>
<td>Ukrainian</td>
</tr>
<tr>
<td>38</td>
<td>Vietnamese</td>
</tr>
<tr>
<td>39</td>
<td>Azeri_Cyrillic_90</td>
</tr>
<tr>
<td>40</td>
<td>Azeri_Latin_90</td>
</tr>
<tr>
<td>41</td>
<td>Chinese_Hong_Kong_Stroke_90</td>
</tr>
</tbody>
</table>
### 2.2.1.2.5 Event Host Type

**Event Host Type** is a 4-byte, signed integer that specifies the type of object used as an event host for an event receiver. The only valid values of the Event Host Type are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>The Event Host Type is invalid.</td>
</tr>
<tr>
<td>0</td>
<td>The event host is a site collection.</td>
</tr>
<tr>
<td>1</td>
<td>The event host is a site (2).</td>
</tr>
<tr>
<td>2</td>
<td>The event host is a list (2).</td>
</tr>
<tr>
<td>3</td>
<td>The event host is a list item.</td>
</tr>
<tr>
<td>4</td>
<td>The event host is a content type.</td>
</tr>
<tr>
<td>5</td>
<td>The event host is a workflow (2).</td>
</tr>
<tr>
<td>6</td>
<td>The event host is a feature.</td>
</tr>
</tbody>
</table>

### 2.2.1.2.6 Event Receiver Type

**Event Receiver Type** is a 32-bit signed integer that specifies the type of an event receiver, which specifies when the handler for the event (2) is invoked. The only valid values of the Event Receiver Type are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The event receiver is invoked before a list item is added.</td>
</tr>
<tr>
<td>2</td>
<td>The event receiver is invoked before a list item is updated.</td>
</tr>
<tr>
<td>3</td>
<td>The event receiver is invoked before a list item is deleted.</td>
</tr>
<tr>
<td>4</td>
<td>The event receiver is invoked before a list item is checked in.</td>
</tr>
<tr>
<td>5</td>
<td>The event receiver is invoked before a list item is checked out.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>6</td>
<td>The event receiver is invoked before a list item checkout is reverted.</td>
</tr>
<tr>
<td>7</td>
<td>The event receiver is invoked before an attachment to a list item is added.</td>
</tr>
<tr>
<td>8</td>
<td>The event receiver is invoked before an attachment to a list item is deleted.</td>
</tr>
<tr>
<td>9</td>
<td>The event receiver is invoked before a document is moved.</td>
</tr>
<tr>
<td>101</td>
<td>The event receiver is invoked before a <strong>field</strong> is added to the schema of a list.</td>
</tr>
<tr>
<td>102</td>
<td>The event receiver is invoked before a field is updated in the schema of a list.</td>
</tr>
<tr>
<td>103</td>
<td>The event receiver is invoked before a field is deleted from the schema of a list.</td>
</tr>
<tr>
<td>104</td>
<td>The event receiver is invoked before a list is added.</td>
</tr>
<tr>
<td>105</td>
<td>The event receiver is invoked before a list is deleted.</td>
</tr>
<tr>
<td>201</td>
<td>The event receiver is invoked before a site collection is deleted.</td>
</tr>
<tr>
<td>202</td>
<td>The event receiver is invoked before a <strong>site (2)</strong> is deleted.</td>
</tr>
<tr>
<td>203</td>
<td>The event receiver is invoked before a site (2) is moved.</td>
</tr>
<tr>
<td>204</td>
<td>The event receiver is invoked before a site (2) is added.</td>
</tr>
<tr>
<td>501</td>
<td>The event receiver is invoked before a workflow (2) is started.</td>
</tr>
<tr>
<td>10001</td>
<td>The event receiver is invoked after a list item is added.</td>
</tr>
<tr>
<td>10002</td>
<td>The event receiver is invoked after a list item is updated.</td>
</tr>
<tr>
<td>10003</td>
<td>The event receiver is invoked after a list item is deleted.</td>
</tr>
<tr>
<td>10004</td>
<td>The event receiver is invoked after a list item is checked in.</td>
</tr>
<tr>
<td>10005</td>
<td>The event receiver is invoked after a list item is checked out.</td>
</tr>
<tr>
<td>10006</td>
<td>The event receiver is invoked after a list item checkout is reverted.</td>
</tr>
<tr>
<td>10007</td>
<td>The event receiver is invoked after an attachment is added to a list item.</td>
</tr>
<tr>
<td>10008</td>
<td>The event receiver is invoked after an attachment is deleted from a list item.</td>
</tr>
<tr>
<td>10009</td>
<td>The event receiver is invoked after a document is moved.</td>
</tr>
<tr>
<td>10010</td>
<td>The event receiver is invoked after a document is transformed by the document transformation feature.</td>
</tr>
<tr>
<td>10101</td>
<td>The event receiver is invoked after a field is added to the schema of a list.</td>
</tr>
<tr>
<td>10102</td>
<td>The event receiver is invoked after a field is updated in the schema of a list.</td>
</tr>
<tr>
<td>10103</td>
<td>The event receiver is invoked after a field is deleted from the schema of a list.</td>
</tr>
<tr>
<td>10104</td>
<td>The event receiver is invoked after a list is added.</td>
</tr>
<tr>
<td>10105</td>
<td>The event receiver is invoked after a list is deleted.</td>
</tr>
<tr>
<td>10201</td>
<td>The event receiver is invoked after a site collection is deleted.</td>
</tr>
<tr>
<td>10202</td>
<td>The event receiver is invoked after a site (2) is deleted.</td>
</tr>
<tr>
<td>10203</td>
<td>The event receiver is invoked after a site (2) is moved.</td>
</tr>
</tbody>
</table>
### 2.2.1.2.7 Excluded Folder Type

**Excluded Folder Type** is a 4-byte integer value that indicates folders that are excluded from common listings of the subfolders in a document library because of their special roles. The only valid values of **Excluded Folder Type** are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No special handling.</td>
</tr>
<tr>
<td>1</td>
<td>Forms folder. This folder holds view pages within the document library or list.</td>
</tr>
<tr>
<td>2</td>
<td>Web images folder. This folder is named &quot;.w&quot; and holds image files in an image library.</td>
</tr>
<tr>
<td>3</td>
<td>Thumbnails folder. This folder is named &quot;.t&quot; and holds thumbnail images in an image library.</td>
</tr>
<tr>
<td>4</td>
<td>The root folder in a list or document library.</td>
</tr>
</tbody>
</table>

### 2.2.1.2.8 LinkDynamic Type

**LinkDynamic** Type is a 1-byte value represented as a single, uppercase ASCII character that tracks various special link types. A **LinkDynamic** type MUST have only one value at a time. A NULL value for **LinkDynamic** is used for a backward link. The only valid non-NULL values of **LinkDynamic** are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>The URL is &quot;static&quot;, which is the default and requires no special handling.</td>
</tr>
<tr>
<td>D</td>
<td>The URL is &quot;dynamic&quot;, which is a link to &lt;Site URL&gt;_vti_bin\shtml.dll\DirName\LeafName. Such links are used to invoke the FrontPage SmartHTML interpreter on a file.</td>
</tr>
<tr>
<td>L</td>
<td>The URL is to a layouts page (that is, it contains a path segment with the string &quot;.layouts&quot;).</td>
</tr>
<tr>
<td>H</td>
<td>The URL is a history link (that is, it contains a path segment with the string &quot;.vti_history&quot;).</td>
</tr>
<tr>
<td>G</td>
<td>A nonabsolute link from an uncustomized document that does not fall into any other category.</td>
</tr>
</tbody>
</table>
2.2.1.2.9 LinkSecurity Type

A LinkSecurity type is a 1-byte value represented as a single, uppercase ASCII character specifying the URI scheme for a link, such as HTTP or HTTPS. A LinkSecurity type MUST have only one value at a time. A NULL value for LinkSecurity is used for a backward link. The only valid non-NULL values of LinkSecurity are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>The URL begins with &quot;http://&quot; (a nonsecure link using the http: scheme).</td>
</tr>
<tr>
<td>T</td>
<td>The URL begins with &quot;shttp://&quot; (an S-HTTP link using the Terisa's shttp: scheme. For more details, see [RFC2660]).</td>
</tr>
<tr>
<td>S</td>
<td>The URL begins with &quot;https://&quot; (an SSL link using the https: or snews: scheme).</td>
</tr>
<tr>
<td>U</td>
<td>The URL is of another unknown scheme.</td>
</tr>
</tbody>
</table>

2.2.1.2.10 LinkType Types

LinkType Types is a 1-byte value represented as a single, uppercase ASCII character; it specifies type information about a link. A LinkType type MUST have only one value at a time. A NULL value for LinkType is used for a backward link. The only valid non-NULL values of a LinkType are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>The link is from the ACTION attribute of a Hypertext Markup Language (HTML) form tag.</td>
</tr>
<tr>
<td>B</td>
<td>The link is from the attribute markup of a bot.</td>
</tr>
<tr>
<td>C</td>
<td>The link is from an autogenerated table of contents. Agents can ignore the link type when determining unreferenced files within a site (2).</td>
</tr>
<tr>
<td>D</td>
<td>The link references programmatic content, as in the HTML OBJECT or APPLET tags.</td>
</tr>
<tr>
<td>E</td>
<td>The link is from a cascading style sheet (CSS).</td>
</tr>
<tr>
<td>F</td>
<td>The link is from the SRC attribute of an HTML FRAME tag.</td>
</tr>
<tr>
<td>G</td>
<td>The link is to a dynamic Web template for the containing document.</td>
</tr>
<tr>
<td>H</td>
<td>The link is from an HTML HREF attribute. This can also be used as a default link type value if a more precise type does not apply.</td>
</tr>
<tr>
<td>I</td>
<td>The link is to a document that the containing document includes via an include bot.</td>
</tr>
<tr>
<td>J</td>
<td>The link is from a field of this list item.</td>
</tr>
<tr>
<td>K</td>
<td>Identical to &quot;H&quot;, except that the link contains an HTML bookmark specifier.</td>
</tr>
<tr>
<td>L</td>
<td>The link is a target in an HTML image map generated from an image map bot.</td>
</tr>
<tr>
<td>M</td>
<td>The link is to an image used in an HTML image map generated from an image map bot.</td>
</tr>
<tr>
<td>O</td>
<td>The link is part of a cross-page Web Part connection.</td>
</tr>
<tr>
<td>P</td>
<td>The link is part of the markup of a Web Part within the source of the containing document.</td>
</tr>
<tr>
<td>Q</td>
<td>The link references a CSS document that provides style information for the containing document.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>R</td>
<td>The link is from the MasterPageFile attribute of the @Page directive in the containing document.</td>
</tr>
<tr>
<td>S</td>
<td>The link is from an HTML SRC attribute.</td>
</tr>
<tr>
<td>T</td>
<td>The link is to the index file used by a text search bot on this page.</td>
</tr>
<tr>
<td>V</td>
<td>The link is based on the properties of the document rather than anything in the document stream. The link type is used in tracking the link between a site (2) and the master page URL used for the site.</td>
</tr>
<tr>
<td>X</td>
<td>The link is from an XML island within an HTML document.</td>
</tr>
<tr>
<td>Y</td>
<td>The link references an HTML document whose HTML BODY tag attributes are used as a template for the attributes of the containing document's BODY tag.</td>
</tr>
<tr>
<td>Z</td>
<td>The link is part of the markup of a Web Part that exists in a Web Part zone in the containing document and is consequently not stored within the source of the containing document.</td>
</tr>
</tbody>
</table>

### 2.2.1.2.11 List Base Type

**List Base Type** is a 32-bit integer enumeration of possible base types for lists. All lists are created with one of these base types, which define implementation-specific common values for list properties. The only valid values of the **List Base Type** are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Generic list</td>
</tr>
<tr>
<td>1</td>
<td>Document library</td>
</tr>
<tr>
<td>3</td>
<td>Discussion board list</td>
</tr>
<tr>
<td>4</td>
<td>Survey list</td>
</tr>
<tr>
<td>5</td>
<td>Issues list</td>
</tr>
</tbody>
</table>

### 2.2.1.2.12 List Server Template

**List Server Template** is a 32-bit integer enumeration of the possible values for the list server template defining the base structure of a list. Reserved values of the **List Server Template** are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>Invalid Template</td>
</tr>
<tr>
<td>100</td>
<td>Generic List Template</td>
</tr>
<tr>
<td>101</td>
<td>Document Library Template</td>
</tr>
<tr>
<td>102</td>
<td>Survey Template</td>
</tr>
<tr>
<td>103</td>
<td>Links Template</td>
</tr>
<tr>
<td>104</td>
<td>Announcements Template</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>105</td>
<td>Contacts Template</td>
</tr>
<tr>
<td>106</td>
<td>Events Template</td>
</tr>
<tr>
<td>107</td>
<td>Tasks Template</td>
</tr>
<tr>
<td>108</td>
<td>Discussion Template</td>
</tr>
<tr>
<td>109</td>
<td>Image Library Template</td>
</tr>
<tr>
<td>110</td>
<td>Data Sources Template</td>
</tr>
<tr>
<td>111</td>
<td>Web Template Catalog Template</td>
</tr>
<tr>
<td>112</td>
<td>User Info Catalog Template</td>
</tr>
<tr>
<td>113</td>
<td>Web Part Catalog Template</td>
</tr>
<tr>
<td>114</td>
<td>List Template Catalog Template</td>
</tr>
<tr>
<td>115</td>
<td>XML Form Template</td>
</tr>
<tr>
<td>116</td>
<td>Master Page Catalog Template</td>
</tr>
<tr>
<td>117</td>
<td>No Code Workflows Template</td>
</tr>
<tr>
<td>118</td>
<td>Workflow Process Template</td>
</tr>
<tr>
<td>119</td>
<td>Webpage Library Template</td>
</tr>
<tr>
<td>120</td>
<td>Custom Grid Template</td>
</tr>
<tr>
<td>121</td>
<td>Solution Catalog Template</td>
</tr>
<tr>
<td>122</td>
<td>No Code Workflows Public Template</td>
</tr>
<tr>
<td>123</td>
<td>Theme Catalog Template</td>
</tr>
<tr>
<td>124</td>
<td>DESIGN Catalog Template</td>
</tr>
<tr>
<td>125</td>
<td>APPDATA Catalog Template</td>
</tr>
<tr>
<td>130</td>
<td>Data Connection Library Template</td>
</tr>
<tr>
<td>140</td>
<td>Workflow History Template</td>
</tr>
<tr>
<td>150</td>
<td>Gantt Tasks Template</td>
</tr>
<tr>
<td>151</td>
<td>Help Library Template</td>
</tr>
<tr>
<td>160</td>
<td>Access Request Template</td>
</tr>
<tr>
<td>171</td>
<td>Tasks with Timeline and Hierarchy Template</td>
</tr>
<tr>
<td>175</td>
<td>Maintenance Logs Template</td>
</tr>
<tr>
<td>200</td>
<td>Meetings Template</td>
</tr>
<tr>
<td>201</td>
<td>Agenda Template</td>
</tr>
<tr>
<td>202</td>
<td>Meeting User Template</td>
</tr>
<tr>
<td>204</td>
<td>Decision (Meeting) Template</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>207</td>
<td>Meeting Objective Template</td>
</tr>
<tr>
<td>210</td>
<td>Textbox Template</td>
</tr>
<tr>
<td>211</td>
<td>Things To Bring (Meeting) Template</td>
</tr>
<tr>
<td>212</td>
<td>Homepage Library Template</td>
</tr>
<tr>
<td>301</td>
<td>Posts (Blog) Template</td>
</tr>
<tr>
<td>302</td>
<td>Comments (Blog) Template</td>
</tr>
<tr>
<td>303</td>
<td>Categories (Blog) Template</td>
</tr>
<tr>
<td>402</td>
<td>Facility Template</td>
</tr>
<tr>
<td>403</td>
<td>Whereabouts Template</td>
</tr>
<tr>
<td>404</td>
<td>Call Track Template</td>
</tr>
<tr>
<td>405</td>
<td>Circulation Template</td>
</tr>
<tr>
<td>420</td>
<td>Timecard Template</td>
</tr>
<tr>
<td>421</td>
<td>Holiday Template</td>
</tr>
<tr>
<td>499</td>
<td>IMEDic Template</td>
</tr>
<tr>
<td>600</td>
<td>External List Template</td>
</tr>
<tr>
<td>700</td>
<td>My Site Document Library Template</td>
</tr>
<tr>
<td>1100</td>
<td>Issue Tracking Template</td>
</tr>
<tr>
<td>1200</td>
<td>Admin Tasks</td>
</tr>
<tr>
<td>1220</td>
<td>Health Rules Template</td>
</tr>
<tr>
<td>1221</td>
<td>Health Reports Template</td>
</tr>
<tr>
<td>1230</td>
<td>Draft Apps Library in Developer Site Template</td>
</tr>
</tbody>
</table>

2.2.1.2.13 Moderation Status

Moderation Status is a 4-byte integer indicating the moderation approval status of a list item. Configurations can require moderation approval to publish a list item or allow automatic approval. A published list item MUST have a Moderation Status of 0. The following are all possible valid values for Moderation Status.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The list item is approved.</td>
</tr>
<tr>
<td>1</td>
<td>The list item has been denied approval.</td>
</tr>
<tr>
<td>2</td>
<td>The list item is pending approval.</td>
</tr>
<tr>
<td>3</td>
<td>The list item is in the draft or checked out state.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>4</td>
<td>The list item is scheduled for automatic approval at a future date.</td>
</tr>
</tbody>
</table>

### 2.2.1.2.14 Page Type

**Page Type** is a signed 1-byte integer that is used to represent the possible page types. The reserved **Page Type** values are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>Does not correspond to a view or a form of a list.</td>
</tr>
<tr>
<td>0</td>
<td>The default view of the corresponding list. This view is displayed whenever this list is viewed without an explicit view being specified.</td>
</tr>
<tr>
<td>1</td>
<td>A view of the corresponding list, but not the default view.</td>
</tr>
<tr>
<td>2</td>
<td>This value is only used internally within implementation-specific code and is never stored in a database.</td>
</tr>
<tr>
<td>3</td>
<td>This value is only used internally within implementation-specific code and is never stored in a database.</td>
</tr>
<tr>
<td>4</td>
<td>A display form of a list, suitable for displaying a single list item in read-only mode.</td>
</tr>
<tr>
<td>5</td>
<td>This value is only used internally within implementation-specific code and is never stored in a database.</td>
</tr>
<tr>
<td>6</td>
<td>An edit form for a list, suitable for presenting UI to update the properties of a list item.</td>
</tr>
<tr>
<td>7</td>
<td>Used to represent edit forms of a list suitable for displaying in HTML file dialogs to a client application.</td>
</tr>
<tr>
<td>8</td>
<td>A new form for a list, suitable for presenting UI to create a new list item.</td>
</tr>
<tr>
<td>9</td>
<td>This value is from a previous implementation and is no longer valid.</td>
</tr>
<tr>
<td>10</td>
<td>This value is only used internally within implementation-specific code and is never stored in a database.</td>
</tr>
</tbody>
</table>

### 2.2.1.2.15 Redirect Type

**Redirect Type** is a 1-byte value corresponding to the type of item that a URL is redirected to. The following table contains all possible values for **Redirect Type**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Welcome page.</td>
</tr>
<tr>
<td>1</td>
<td>Homepage.</td>
</tr>
<tr>
<td>2</td>
<td>List view - redirect to root folder.</td>
</tr>
<tr>
<td>3</td>
<td>Provision - redirect to a template picker page during provisioning of a new list.</td>
</tr>
<tr>
<td>255</td>
<td>None.</td>
</tr>
</tbody>
</table>

### 2.2.1.2.16 Role Definition Type
**Role Definition Type** is a 1-byte value that is used to represent the type of implementation-specific default and custom Role Definitions. This integer value MUST be a value enumerated in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A custom-defined role</td>
</tr>
<tr>
<td>1</td>
<td>Guest</td>
</tr>
<tr>
<td>2</td>
<td>Reader</td>
</tr>
<tr>
<td>3</td>
<td>Contributor</td>
</tr>
<tr>
<td>4</td>
<td>Web Designer</td>
</tr>
<tr>
<td>5</td>
<td>Administrator</td>
</tr>
</tbody>
</table>

**2.2.1.2.17 Virus Status**

**Virus Status** is a 4-byte, integer enumerated type that specifies the current virus scan status of a document. The following are valid values for *Virus Status*.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>This document is reported as clean from viruses.</td>
</tr>
<tr>
<td>1</td>
<td>This document had a virus reported by the virus scanner plug-in.</td>
</tr>
<tr>
<td>2</td>
<td>This document had a virus reported by the virus scanner plug-in, which the scanner determines that it can remove.</td>
</tr>
<tr>
<td>3</td>
<td>This document had a virus previously reported, but the virus scanner determines that it successfully removed it.</td>
</tr>
<tr>
<td>4</td>
<td>This document had a virus reported, and the virus scanner attempted to clean it but failed.</td>
</tr>
<tr>
<td>5</td>
<td>This document had a virus reported, and the scanner requested that the document be deleted.</td>
</tr>
<tr>
<td>6</td>
<td>This document had a timeout from the virus scanner when it was last processed.</td>
</tr>
</tbody>
</table>

**2.2.1.2.18 App Principal Permissions Enumeration**

The following table lists the values and the descriptions for the *App Principal Permissions* enumeration.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No permissions.</td>
</tr>
<tr>
<td>1</td>
<td>Guest permissions.</td>
</tr>
<tr>
<td>2</td>
<td>Read permissions.</td>
</tr>
<tr>
<td>3</td>
<td>Write permissions.</td>
</tr>
<tr>
<td>4</td>
<td>Manage permissions.</td>
</tr>
</tbody>
</table>
2.2.1.3 Time Zone Identifier

Time Zone Identifier is a 2-byte integer value identifying a time zone. The only valid values of the Time Zone Identifier are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>(GMT+01:00) Brussels, Copenhagen, Madrid, Paris</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(GMT+02:00) Athens, Bucharest, Istanbul</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(GMT+02:00) Minsk</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(GMT-03:00) Brasilia</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(GMT-04:00) Atlantic Time (Canada)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(GMT-05:00) Eastern Time (U.S. and Canada)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(GMT-06:00) Central Time (U.S. and Canada)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(GMT-07:00) Mountain Time (U.S. and Canada)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>(GMT-08:00) Pacific Time (U.S. and Canada)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>(GMT-09:00) Alaska</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>(GMT-10:00) Hawaii</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(GMT-11:00) Midway Island, Samoa</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>(GMT+12:00) Auckland, Wellington</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>(GMT+10:00) Brisbane</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>(GMT+09:30) Adelaide</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>(GMT+09:00) Osaka, Sapporo, Tokyo</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>(GMT+08:00) Kuala Lumpur, Singapore</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>(GMT+07:00) Bangkok, Hanoi, Jakarta</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>(GMT+04:00) Abu Dhabi, Muscat</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>(GMT+03:30) Tehran</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>(GMT+03:00) Baghdad</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>(GMT+02:00) Jerusalem</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>(GMT-03:30) Newfoundland</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>(GMT-01:00) Azores</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>(GMT-02:00) Mid-Atlantic</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>(GMT) Casablanca, Monrovia, Reykjavik</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>(GMT-03:00) Buenos Aires, Georgetown</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>(GMT-04:00) Caracas, La Paz</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>(GMT-05:00) Indiana (East)</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>(GMT-05:00) Bogota, Lima, Quito, Rio Branco</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>(GMT-06:00) Saskatchewan</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>(GMT-06:00) Guadalajara, Mexico City, Monterrey</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>(GMT-07:00) Arizona</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>(GMT-12:00) International Date Line West</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>(GMT+12:00) Fiji Is., Kamchatka, Marshall Is.</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>(GMT+11:00) Magadan, Solomon Is., New Caledonia</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>(GMT+10:00) Hobart</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>(GMT+10:00) Guam, Port Moresby</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>(GMT+09:30) Darwin</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>(GMT+08:00) Beijing, Chongqing, Hong Kong S.A.R., Urumqi</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>(GMT+06:00) Almaty, Novosibirsk</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>(GMT+05:00) Islamabad, Karachi, Tashkent</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>(GMT+04:30) Kabul</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>(GMT+02:00) Cairo</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>(GMT+02:00) Harare, Pretoria</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>(GMT+03:00) Moscow, St. Petersburg, Volgograd</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>(GMT-01:00) Cape Verde Is.</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>(GMT+04:00) Baku</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>(GMT-06:00) Central America</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>(GMT+03:00) Nairobi</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>(GMT+05:00) Ekaterinburg</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>(GMT-03:00) Greenland</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>(GMT+06:30) Yangon (Rangoon)</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>(GMT+05:45) Kathmandu</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>(GMT+08:00) Irkutsk, Ulaan Bataar</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>(GMT+07:00) Krasnoyarsk</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>(GMT-04:00) Santiago</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>(GMT+05:30) Sri Jayawardenepura</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>(GMT+13:00) Nuku'alofa</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>(GMT+10:00) Vladivostok</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>(GMT+01:00) West Central Africa</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>(GMT+09:00) Yakutsk</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>(GMT+06:00) Astana, Dhaka</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>(GMT+09:00) Seoul</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>(GMT+08:00) Perth</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>(GMT+03:00) Kuwait, Riyadh</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>(GMT+08:00) Taipei</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>(GMT+10:00) Canberra, Melbourne, Sydney</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>(GMT-07:00) Chihuahua, La Paz, Mazatlan</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>(GMT-08:00) Tijuana, Baja California</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>(GMT+02:00) Amman</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>(GMT+02:00) Beirut</td>
<td></td>
</tr>
</tbody>
</table>
### 2.2.2 Bit Fields and Flag Structures

#### 2.2.2.1 Audit Flags

Audit Flags is a 4-byte unsigned integer bit mask that tracks operations to be audited on a given object. Auditing is an implementation-specific capability, which can have one or more flags set. The values of the Audit Flags bits are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>Audit checkout operations.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Audit checkin operations.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>Audit view operations.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>Audit delete operations.</td>
</tr>
<tr>
<td>0x00000010</td>
<td>Audit update operations.</td>
</tr>
<tr>
<td>0x00000020</td>
<td>Audit content type update operations.</td>
</tr>
<tr>
<td>0x00000040</td>
<td>Audit child object deletion operations.</td>
</tr>
</tbody>
</table>
## 2.2.2.2 Configuration Object Status

**Configuration Object Status** is a 4-byte unsigned integer that describes the status of the associated Configuration Object. Valid values of the **Configuration Object Status** bits are specified in the following table. The semantic meaning of each value is implementation-specific to the service that owns the Configuration Object.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>The Configuration Object is provisioned and online.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>The Configuration Object is disabled. The resources necessary to run this Configuration Object are on the machine, but the administrator needs to provision the Configuration Object and enable it.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>The Configuration Object is offline for some unknown reason.</td>
</tr>
<tr>
<td>0x00000003</td>
<td>The administrator has issued the command to unprovision the Configuration Object into a disabled state, but the unprovisioning job has not completed yet.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>The administrator has issued the command to provision the Configuration Object and turn it online, but the provisioning job has not completed yet.</td>
</tr>
<tr>
<td>0x00000005</td>
<td>The administrator has issued the command to upgrade the Configuration Object into an online state, but the upgrade job has not completed yet.</td>
</tr>
</tbody>
</table>

## 2.2.2.3 Doc Flags

**Doc Flags** is a 4-byte unsigned integer bit mask that provides metadata about the document, which can have one or more flags set. The values of the **Doc Flags** bits are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>This document contains dynamic content to be sent through the CAML interpreter, an implementation-specific dynamic content generation component. An example of this would be a category <strong>Web bot</strong> present in the source of the page.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>The document is a &quot;sub image&quot; of another document. This is set if this document is an automatically generated thumbnail or Web image based on another item in the store.</td>
</tr>
</tbody>
</table>
### 2.2.2.4 Document Store Type

The **Document Store Type** is a 1-byte unsigned integer value that specifies the type of a document or the target of a link within or to a document. The only valid values for a **Document Store Type** are specified as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>File</td>
</tr>
</tbody>
</table>
### 2.2.2.5 List Flags

**List Flags** is an 8-byte unsigned integer bit mask that provides metadata about the list, which can have one or more flags set. **List Flags** identify an implementation-specific capability. The values of the **List Flags** bits are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>Folder</td>
</tr>
<tr>
<td>0x02</td>
<td>Site</td>
</tr>
<tr>
<td>0x80</td>
<td>Backward Link</td>
</tr>
<tr>
<td></td>
<td>Backward Link can only be a return value and MUST NOT be used as an input</td>
</tr>
<tr>
<td></td>
<td>value for a stored procedure.</td>
</tr>
<tr>
<td>0x0000000000000001</td>
<td>This list is an &quot;ordered list&quot;.</td>
</tr>
<tr>
<td>0x0000000000000002</td>
<td>This bit MUST be ignored.</td>
</tr>
<tr>
<td>0x0000000000000004</td>
<td>This list is &quot;undeletable&quot; (that is, it is crucial to the functioning of the</td>
</tr>
<tr>
<td></td>
<td>containing site (2) or site collection).</td>
</tr>
<tr>
<td>0x0000000000000008</td>
<td>Attachments on list items are disabled. This MUST be set if the list is</td>
</tr>
<tr>
<td></td>
<td>a document library or survey.</td>
</tr>
<tr>
<td>0x0000000000000010</td>
<td>This list is a &quot;catalog&quot; (for example, a Web Part gallery or master page</td>
</tr>
<tr>
<td></td>
<td>gallery).</td>
</tr>
<tr>
<td>0x0000000000000020</td>
<td>This list is associated with a site (2) using the meetings workspace site</td>
</tr>
<tr>
<td></td>
<td>template and contains data scoped to each instance of a recurring meeting.</td>
</tr>
<tr>
<td>0x0000000000000040</td>
<td>This list generates alerts when a list item is assigned to a user.</td>
</tr>
<tr>
<td>0x0000000000000080</td>
<td>This list has versioning enabled, and supports creating historical versions</td>
</tr>
<tr>
<td></td>
<td>of list items when changes occur. This bit MUST be ignored for Lists with</td>
</tr>
<tr>
<td></td>
<td>a List Base Type of survey.</td>
</tr>
<tr>
<td>0x0000000000000100</td>
<td>This list MUST be hidden from enumeration functions. This is intended for</td>
</tr>
<tr>
<td></td>
<td>lists implementing infrastructure for an application.</td>
</tr>
<tr>
<td>0x0000000000000200</td>
<td>This list is configured to bring up a page to fill out a form to request</td>
</tr>
<tr>
<td></td>
<td>access from the owner when a user is denied access while browsing its list</td>
</tr>
<tr>
<td></td>
<td>items.</td>
</tr>
<tr>
<td>0x0000000000000400</td>
<td>This list has moderation enabled, requiring an approval process when</td>
</tr>
<tr>
<td></td>
<td>content is created or modified.</td>
</tr>
<tr>
<td>0x0000000000000800</td>
<td>If this list is a survey, it will allow multiple responses for a given</td>
</tr>
<tr>
<td></td>
<td>user rather than restricting users to a single response. This flag MUST</td>
</tr>
<tr>
<td></td>
<td>be ignored for lists that do not have a List Base Type of survey.</td>
</tr>
<tr>
<td>0x0000000000001000</td>
<td>This list uses the value of each field's ForcedDisplay attribute when</td>
</tr>
<tr>
<td></td>
<td>presenting data from that field. This is commonly used in anonymous</td>
</tr>
<tr>
<td></td>
<td>surveys to display common placeholder text wherever the respondent's name</td>
</tr>
<tr>
<td></td>
<td>would normally appear.</td>
</tr>
<tr>
<td>0x0000000000002000</td>
<td>This list MUST NOT be serialized as part of saving this site (2) as a site</td>
</tr>
<tr>
<td></td>
<td>template.</td>
</tr>
<tr>
<td>0x0000000000004000</td>
<td>The List Server Template for this list can only be instantiated in the</td>
</tr>
<tr>
<td></td>
<td>root site (2) of a given site collection.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x000000000000080000</td>
<td>When a <strong>List Server Template</strong> is being created for this list, documents in the root of the list can also be serialized.</td>
</tr>
<tr>
<td>0x0000000000010000</td>
<td>Insertion of list items via email is enabled for this list.</td>
</tr>
<tr>
<td>0x0000000000020000</td>
<td>This is a &quot;private&quot; list. When a <strong>List Server Template</strong> based on this list is created, the new list can be given an <strong>ACL</strong> so that only its owner and administrators can access the list.</td>
</tr>
<tr>
<td>0x0000000000040000</td>
<td>This document library requires the user to check out documents before modifying them.</td>
</tr>
<tr>
<td>0x0000000000080000</td>
<td>This list supports creation of <strong>minor versions</strong> on item revisions.</td>
</tr>
<tr>
<td>0x0000000000100000</td>
<td>This document library requires that users have the <strong>EditListItems</strong> right to see minor versions of documents.</td>
</tr>
<tr>
<td>0x0000000000200000</td>
<td>This document library requires that users have the <strong>ApproveItems</strong> right to see minor versions of documents.</td>
</tr>
<tr>
<td>0x0000000000400000</td>
<td>This list supports displaying a user interface for manipulating multiple content types (for example, a list that contains both announcements and tasks).</td>
</tr>
<tr>
<td>0x0000000000800000</td>
<td>This list has had its schema customized from the version that exists in the on-disk schema file that was used to create it.</td>
</tr>
<tr>
<td>0x0000000001000000</td>
<td>Document parsers in this list generate thumbnail files corresponding to documents saved to this list. This bit MUST be ignored for lists that are not document libraries.</td>
</tr>
<tr>
<td>0x0000000002000000</td>
<td>Documents or list item attachments in this list can be directly browsed to by anyone who has access to the list itself. This is useful for shared resources such as the master page gallery, where one page can be used throughout a site collection in scopes with varying permissions.</td>
</tr>
<tr>
<td>0x0000000004000000</td>
<td>This list has workflows (2) associated with it.</td>
</tr>
<tr>
<td>0x0000000008000000</td>
<td>This list MUST NOT be automatically exported when exporting a list that references it. Exporting is an implementation-specific capability.</td>
</tr>
<tr>
<td>0x0000000001000000</td>
<td>Applications generating server transformations of list items in this list SHOULD default to open the list item in a browser rather than in a separate client-side application. Server transformations are performed by server-side document viewers that can allow clients to view documents without additional client software. Server transformations are an implementation-specific feature.</td>
</tr>
<tr>
<td>0x0000000002000000</td>
<td>Creation of folders is blocked in this list.</td>
</tr>
<tr>
<td>0x0000000004000000</td>
<td>This list disallows advanced view functionality, such as the <strong>datasheet</strong> view and views involving Web Part to Web Part connections.</td>
</tr>
<tr>
<td>0x0000000008000000</td>
<td>This list specifies custom sorting orders for the list of content types available on a per-folder basis.</td>
</tr>
<tr>
<td>0x0000000010000000</td>
<td>This list MUST NOT be exported as part of a migration package. Migration packages are an implementation-specific feature.</td>
</tr>
<tr>
<td>0x0000000020000000</td>
<td>This list has its schema cached in memory when possible, rather than retrieving the schema every time the list is accessed.</td>
</tr>
<tr>
<td>0x0000000040000000</td>
<td>This bit MUST NOT be returned by the database.</td>
</tr>
<tr>
<td>0x0000000080000000</td>
<td>This list's content is not processed by a search <strong>crawler</strong>.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x000000010000000000</td>
<td>Data from this list MUST be included when it is saved as a <strong>List Server Template</strong>, even if not otherwise requested.</td>
</tr>
<tr>
<td>0x000000020000000000</td>
<td>Content type manipulation is disabled on this list.</td>
</tr>
<tr>
<td>0x000000040000000000</td>
<td><strong>RSS</strong> feed syndication is disabled for this list.</td>
</tr>
<tr>
<td>0x000000080000000000</td>
<td>Information Rights Management (IRM) is enabled for documents in this list.</td>
</tr>
<tr>
<td>0x000001000000000000</td>
<td>Expiration of IRM rights is enabled for this list. Setting this bit requires that the IRM enabled bit also be set.</td>
</tr>
<tr>
<td>0x000002000000000000</td>
<td>Documents that do not have a registered IRM protector will be blocked from this list.</td>
</tr>
<tr>
<td>0x000008000000000000</td>
<td>If this list is an events list, this list supports a user interface which allows the user to select user identities when adding new list items.</td>
</tr>
<tr>
<td>0x000010000000000000</td>
<td>This list has data validation criteria used to perform custom validation rules prior to the list being updated.</td>
</tr>
<tr>
<td>0x000020000000000000</td>
<td>If this list is an events list, this list supports a user interface for adding resources.</td>
</tr>
<tr>
<td>0x000040000000000000</td>
<td>The data for this list comes from an external data source.</td>
</tr>
<tr>
<td>0x000080000000000000</td>
<td>Calculated expressions in this list MUST preserve NULL values rather than converting them to empty strings.</td>
</tr>
<tr>
<td>0x000100000000000000</td>
<td>This list has list scoped <strong>custom actions</strong>.</td>
</tr>
<tr>
<td>0x000020000000000000</td>
<td>This site (2) MUST NOT be taken offline by a client application.</td>
</tr>
<tr>
<td>0x000400000000000000</td>
<td>Protocol clients MUST enforce validation rules defined for this list. Validation rules are an implementation-specific capability.</td>
</tr>
<tr>
<td>0x000800000000000000</td>
<td>The flag for this list with value 0x0000000010000000 MUST override any server-wide settings for server transformations.</td>
</tr>
<tr>
<td>0x001000000000000000</td>
<td>This list is part of the infrastructure of the site (2) that contains it and client applications MUST treat it as a top level navigation object when displaying information about the site that contains it.</td>
</tr>
<tr>
<td>0x002000000000000000</td>
<td>Views of this list MUST NOT display options to bulk edit data in a datasheet view.</td>
</tr>
<tr>
<td>0x004000000000000000</td>
<td>Protocol clients SHOULD prevent browser execution of active content from files in this library.</td>
</tr>
<tr>
<td>0x008000000000000000</td>
<td>The user interface for forms in this list SHOULD navigate to form pages instead of hosting the list form in a modal dialog.</td>
</tr>
<tr>
<td>0x010000000000000000</td>
<td>Calculated expressions in this list MUST return error values in the case of data type mismatch.</td>
</tr>
<tr>
<td>0x020000000000000000</td>
<td>This list at one time supported creation of minor versions on item revisions, so minor versions might exist for items in this list.</td>
</tr>
<tr>
<td>0xFC00000000000000</td>
<td>Unused.</td>
</tr>
</tbody>
</table>
2.2.2.6 Publishing Level Type

A Publishing Level Type is an 8-bit unsigned integer that describes the publishing level of a version of a document. Only the following values are valid for the Publishing Level Type.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The default value, referring to a published document shown to all users who have access to its storage location.</td>
</tr>
<tr>
<td>2</td>
<td>This document is in a draft state and is shown only to users who have permissions to see documents in a draft state.</td>
</tr>
<tr>
<td>255</td>
<td>This document is checked out to a particular user and shown only to that user.</td>
</tr>
</tbody>
</table>

2.2.2.7 Put Flags Type

Put Flags Type is a 4-byte integer bit mask containing option flags for adding or updating a document. Zero or more of the following bit flags can be set in a Put Flags Type. The values of the PutFlags bits are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Unconditionally update the document.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>Keep the document checked out.</td>
</tr>
<tr>
<td>0x00000010</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000020</td>
<td>Check the document in.</td>
</tr>
<tr>
<td>0x00000040</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000080</td>
<td>Create a thicket or thicket supporting folders.</td>
</tr>
<tr>
<td>0x00000100</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000200</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000400</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000800</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00001000</td>
<td>Create a new user interface (UI) version of the document, even if it is in a short-term lock.</td>
</tr>
<tr>
<td>0x00002000</td>
<td>Set only when a document migration operation is occurring.</td>
</tr>
<tr>
<td>0x00004000</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00008000</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00010000</td>
<td>Publish the document: change user interface (UI) version to published.</td>
</tr>
<tr>
<td>0x00020000</td>
<td>Overwrite the document without updating the displayed version of the document.</td>
</tr>
<tr>
<td>0x00040000</td>
<td>Unpublish document: Change user interface (UI) version from published to draft.</td>
</tr>
</tbody>
</table>
### 2.2.2.8 Rename Flags

**Rename Flags** is a 4-byte integer bit mask containing option flags for renaming a document. This bit mask can have zero or more flags set. The reserved values of the **renameFlags** bits are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Default behavior: Rename all dependent items.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Do not update all related documents. [2]</td>
</tr>
<tr>
<td>0x00000002</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>Server SHOULD find backward links to update other documents linking to the original document.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000010</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00000020</td>
<td>Allow renaming of sites.</td>
</tr>
<tr>
<td>0x00000040</td>
<td>Set the <strong>Doc Flags</strong> (section 2.2.3) bit 0x00000004 to match this value when a file's extension changes.</td>
</tr>
<tr>
<td>0x00000080</td>
<td>Set the <strong>Doc Flags</strong> bit 0x00000008 to match this value when a file's extension changes.</td>
</tr>
<tr>
<td>0x00000100</td>
<td>This value is only used internally within implementation-specific code.</td>
</tr>
<tr>
<td>0x00000200</td>
<td>Allow move into the <strong>forms</strong> directory.</td>
</tr>
<tr>
<td>0x00000400</td>
<td>Current user can view draft versions of the source documents in a move operation.</td>
</tr>
<tr>
<td>0x00000800</td>
<td>Allow rename operation on a thicket main file with missing thicket supporting files.</td>
</tr>
<tr>
<td>0x00001000</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>0x00002000</td>
<td>Overwrite existing files in the destination if the source file is newer.</td>
</tr>
<tr>
<td>0x00004000</td>
<td>MUST be ignored by the back-end database server.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0xFFFF8000</td>
<td>Unused.</td>
</tr>
</tbody>
</table>

### 2.2.2.9 Site Collection Flags

**Site Collection Flags** is a 4-byte, unsigned integer bit mask that specifies properties that are global to a site collection. This bit mask can have zero or more flags set. The values of the **Site Collection Flags** bits are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>This site collection has been write-locked, and user write operations will be blocked.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>This site collection is fully locked, and user read and write operations will be blocked.</td>
</tr>
<tr>
<td>0x000003FC</td>
<td>Unused.</td>
</tr>
<tr>
<td>0x00000400</td>
<td>The site collection has sent a notification that disk usage is near its limit.</td>
</tr>
<tr>
<td>0x00000800</td>
<td>Unused.</td>
</tr>
<tr>
<td>0x00001000</td>
<td>The number of users in this site collection is large. Special consideration can be given to query execution plans when retrieving data.</td>
</tr>
<tr>
<td>0x00020000</td>
<td>The site collection has disabled syndication via RSS.</td>
</tr>
<tr>
<td>0x00040000</td>
<td>Unused.</td>
</tr>
<tr>
<td>0x00080000</td>
<td>The site collection is restricted to users with user accounts from a particular directory path within Active Directory.</td>
</tr>
<tr>
<td>0x00100000</td>
<td>There are sandboxed solutions activated in the site collection.</td>
</tr>
<tr>
<td>0x00200000</td>
<td>Email has been sent out to site collection owner about the resource usage for sandboxed solutions exceeding the resource quota warning level.</td>
</tr>
<tr>
<td>0x00400000</td>
<td>Email has been sent out to the site collection owner about resource usage for sandboxed solutions exceeding the resource quota limit.</td>
</tr>
<tr>
<td>0x00800000</td>
<td>Resource usage for the sandbox solutions exceeding the resource quota limit.</td>
</tr>
<tr>
<td>0x01000000</td>
<td>This site collection has site collection scoped custom actions.</td>
</tr>
<tr>
<td>0x02000000</td>
<td>The visual Upgrade site collection UI is shown. Visual Upgrade feature allows users to choose the UI of upgraded sites. If the user chooses to update, the sites will use the new interface, which includes the ribbon; otherwise, the sites will continue to use Windows SharePoint Services 3.0 UI.</td>
</tr>
<tr>
<td>0x04000000</td>
<td>The visual Upgrade Site Setting UI is shown on sites in this Site Collection. Visual Upgrade feature allows users to choose the UI of upgraded sites. If the user chooses to update, the sites will use the new interface, which includes the ribbon; otherwise, the sites will continue to use Windows SharePoint Services 3.0 UI.</td>
</tr>
</tbody>
</table>
### Site Collection Upgrade Flags

**Site Collection Upgrade Flags** is a 4-byte, unsigned integer bit mask that specifies properties that are global to a site collection regarding site collection upgrade. This bit mask can have zero or more flags set. The values of the **Site Collection Upgrade Flags** bits are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>Whether site collection upgrade is allowed on this site collection is controlled by the parent of the site collection.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Site collection upgrade is allowed on this site collection. This bit is ignored if bit 0x00000001 is set.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>The site collection upgrade is in progress for this site collection.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>Whether upgrade evaluation site collection is allowed to be created is controlled at this site collection. When this bit is set, use the setting at the parent of the site collection.</td>
</tr>
<tr>
<td>0x00000010</td>
<td>Upgrade evaluation site collection is allowed to be created. This bit is ignored if bit 0x00000008 is not set.</td>
</tr>
</tbody>
</table>

### Site Property Flags

**Site Property Flags** is a 4-byte, unsigned integer bit mask that tracks property flags applied to a site (2). The site (2) can have one or more **Site Property Flags** set. These flags reference implementation-specific capabilities. The values of the **Site Property Flags** bits are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>This site (2) allows display of implementation-specific user presence information in the front-end Web server.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>This site (2) allows display of implementation-specific enhanced user presence information in the front-end Web server.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>HTML views for file dialogs MUST NOT be displayed for this site (2).</td>
</tr>
<tr>
<td>0x00000008</td>
<td>This site (2) has disabled syndication of list items via RSS.</td>
</tr>
<tr>
<td>0x00000010</td>
<td>Unused.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x00000020</td>
<td>The front-end Web server for this site (2) displays the &quot;quick launch&quot; navigational element.</td>
</tr>
<tr>
<td>0x00000040</td>
<td>The front-end Web server for this site (2) displays a hierarchical &quot;tree view&quot; navigational element.</td>
</tr>
<tr>
<td>0x00000080</td>
<td>Document parsing is disabled for this site (2).</td>
</tr>
<tr>
<td>0x00000100</td>
<td>This site (2) has not yet been provisioned with a site template.</td>
</tr>
<tr>
<td>0x00000200</td>
<td>List schema information can be cached for lists within this site (2).</td>
</tr>
<tr>
<td>0x00000400</td>
<td>This site (2) has at least one uniquely secured object within it.</td>
</tr>
<tr>
<td>0x00000800</td>
<td>Search indexing agents can index the rendered content from ASPX pages within this site (2).</td>
</tr>
<tr>
<td>0x00010000</td>
<td>The site (2) has at least one custom action that resides at the site level.</td>
</tr>
<tr>
<td>0x00020000</td>
<td>The site (2) has at least one custom action that resides at the list level.</td>
</tr>
<tr>
<td>0x00040000</td>
<td>This site (2) has been initialized since upgrade occurred.</td>
</tr>
<tr>
<td>0x00080000</td>
<td>This site (2) isn't taken offline by the client.</td>
</tr>
<tr>
<td>0xFFFFE000</td>
<td>Unused.</td>
</tr>
</tbody>
</table>

### 2.2.2.12 UserInfo tp_flags

A 4-byte integer bit mask determining the user's options. This can have one or more flags set. The default value is 0, but it MUST NOT be NULL. The valid flags are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>None</td>
</tr>
<tr>
<td>0x00000001</td>
<td>The user is associated with an external application.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>When the user accesses the front-end Web server, the request MUST have an X-RequestToken HTTP Header, which is an opaque string value generated by the front-end Web server.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>The user will not have permission to browse the user information list.</td>
</tr>
</tbody>
</table>

### 2.2.2.13 View Flags

**View Flags** is a 4-byte, integer bit mask that corresponds to properties of a view. This bit mask can have zero or more flags set. The values of the **View Flags** bits are specified as follows.
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>Normal HTML-based view.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>View has been modified by a client application such that it isn't compatible with the Web interface for view modification. Implementations MUST restrict modifying any properties they do not understand.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>Tabular view. The view will render a check box in front of each row to use for bulk updates.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>View MUST NOT be displayed in enumerations of the views of this List (that is, in a view selector front-end Web server element).</td>
</tr>
<tr>
<td>0x00000010</td>
<td>Value is unused and MUST be ignored.</td>
</tr>
<tr>
<td>0x00000020</td>
<td>View is read-only, and implementations MUST NOT allow any modifications to its properties.</td>
</tr>
<tr>
<td>0x00000040</td>
<td>If the query for this view returns no rows, implementations of the front-end Web server MUST return an HTTP 410 error when displaying this view as part of an HTTP request, instead of displaying a normal empty view body.</td>
</tr>
<tr>
<td>0x00000080</td>
<td>Presents data in a nontabular fashion. Implementations can format results in a manner compatible with free-form presentation.</td>
</tr>
<tr>
<td>0x00000100</td>
<td>View suitable for displaying in an HTML-based file navigation dialog to client applications.</td>
</tr>
<tr>
<td>0x00000200</td>
<td>Value is unused, and implementations MUST ignore it.</td>
</tr>
<tr>
<td>0x00000400</td>
<td>View has functionality for aggregating data across multiple XML documents within an XML form library.</td>
</tr>
<tr>
<td>0x00000800</td>
<td>View presents a datasheet view to a rich client application.</td>
</tr>
<tr>
<td>0x00001000</td>
<td>Displays all items in the list recursively from the specified folder, instead of only displaying the immediate children of the current folder.</td>
</tr>
<tr>
<td>0x00002000</td>
<td>Requires that view's data be expanded based on a calendar recurrence (for example, having a view of list item data for the first Thursday of every month).</td>
</tr>
<tr>
<td>0x00004000</td>
<td>View is the system-created view of a user's items awaiting moderation in a moderated list.</td>
</tr>
<tr>
<td>0x00008000</td>
<td>System-created moderator's view of a moderated list, which displays list items pending approval.</td>
</tr>
<tr>
<td>0x00010000</td>
<td>Threaded view for legacy discussion boards (lists with base type 3). Implementations MUST display results in a threaded fashion, and paging of results MUST be done in terms of threads instead of by individual list items.</td>
</tr>
<tr>
<td>0x00020000</td>
<td>Displays HTML-based graphical charts of list item data.</td>
</tr>
<tr>
<td>0x00040000</td>
<td>A personal view, which MUST only be displayed to the user who created the view.</td>
</tr>
<tr>
<td>0x00080000</td>
<td>Displays data on a calendar based on date and time properties of the list items.</td>
</tr>
<tr>
<td>0x00100000</td>
<td>Default form of the specified type for the corresponding list.</td>
</tr>
<tr>
<td>0x00200000</td>
<td>Does not display any list items that are folders.</td>
</tr>
<tr>
<td>0x00400000</td>
<td>Displays list items based on the item order of the list. If this list is not an ordered list, this value MUST be 0.</td>
</tr>
<tr>
<td>0x00800000</td>
<td>View intended for display on mobile devices.</td>
</tr>
<tr>
<td>0x01000000</td>
<td>View is displayed as the default view of this list when a mobile view is requested.</td>
</tr>
<tr>
<td>0x02000000</td>
<td>Displays historical versions of list items.</td>
</tr>
</tbody>
</table>
### Workdays Flag

Workdays Flag is a 2-byte, bit mask that is used to specify the workdays in a week for display in a calendar. The workdays are specified as the sum or bitwise OR of the bit flags specifying each day of the week. For example, if the first day of the week is a Sunday and the workdays are Monday through Friday, the combined flags value would be 0x003E, or 62. The only valid values of the Workdays Flag bits are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0001</td>
<td>Saturday is a workday.</td>
</tr>
<tr>
<td>0x0002</td>
<td>Friday is a workday.</td>
</tr>
<tr>
<td>0x0004</td>
<td>Thursday is a workday.</td>
</tr>
<tr>
<td>0x0008</td>
<td>Wednesday is a workday.</td>
</tr>
<tr>
<td>0x0010</td>
<td>Tuesday is a workday.</td>
</tr>
<tr>
<td>0x0020</td>
<td>Monday is a workday.</td>
</tr>
<tr>
<td>0x0040</td>
<td>Sunday is a workday.</td>
</tr>
<tr>
<td>0xFF80</td>
<td>Unused and MUST NOT be set.</td>
</tr>
</tbody>
</table>

### WSS Rights Mask

WSS Rights Mask is an 8-byte, unsigned integer that specifies the rights that can be assigned to a user or site group. This bit mask can have zero or more flags set.

The values of the permission mask bits are specified as follows.

<table>
<thead>
<tr>
<th>Symbolic name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmptyMask</td>
<td>0x0000000000000000</td>
<td>Grant no permissions.</td>
</tr>
<tr>
<td>FullMask</td>
<td>0x7FFFFFFFFFFFFFFF</td>
<td>Grant all permissions.</td>
</tr>
</tbody>
</table>

The list and document permissions (0x0000000000000000) are specified as follows.
<table>
<thead>
<tr>
<th>Symbolic name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViewListItems</td>
<td>0x0000000000000001</td>
<td>Allow viewing of list items in lists, documents in document libraries, and Web discussion comments.</td>
</tr>
<tr>
<td>AddListItems</td>
<td>0x0000000000000002</td>
<td>Allow addition of list items to lists, documents to document libraries, and Web discussion comments.</td>
</tr>
<tr>
<td>EditListItems</td>
<td>0x0000000000000004</td>
<td>Allow editing of list items in lists, documents in document libraries, Web discussion comments, and to customize Web Part pages in document libraries.</td>
</tr>
<tr>
<td>DeleteListItems</td>
<td>0x0000000000000008</td>
<td>Allow deletion of list items from lists, documents from document libraries, and Web discussion comments.</td>
</tr>
<tr>
<td>ApproveItems</td>
<td>0x0000000000000010</td>
<td>Allow approval of minor versions of a list item or document.</td>
</tr>
<tr>
<td>OpenItems</td>
<td>0x0000000000000020</td>
<td>Allow viewing the source of documents with server-side file handlers.</td>
</tr>
<tr>
<td>ViewVersions</td>
<td>0x0000000000000040</td>
<td>Allow viewing of past versions of a list item or document.</td>
</tr>
<tr>
<td>DeleteVersions</td>
<td>0x0000000000000080</td>
<td>Allow deletion of past versions of a list item or document.</td>
</tr>
<tr>
<td>CancelCheckout</td>
<td>0x0000000000000100</td>
<td>Allow discard or check in of a document that is checked out to another user.</td>
</tr>
<tr>
<td>ManagePersonalViews</td>
<td>0x0000000000000200</td>
<td>Allow creation, change, and deletion of personal views of lists.</td>
</tr>
<tr>
<td></td>
<td>0x0000000000000400</td>
<td>Reserved.</td>
</tr>
<tr>
<td>ManageLists</td>
<td>0x0000000000000800</td>
<td>Allow creation and deletion of lists, addition or removal of fields to the schema of a list, and addition or removal of personal views of a list.</td>
</tr>
<tr>
<td>ViewFormPages</td>
<td>0x0000000000001000</td>
<td>Allow viewing of forms, views, and application pages, and enumerate lists.</td>
</tr>
<tr>
<td>AnonymousSearchAccessList</td>
<td>0x0000000000002000</td>
<td>Allow anonymous users to retrieve content of a list or document library through SharePoint search. The list permissions in the site (2) do not change.</td>
</tr>
<tr>
<td></td>
<td>0x000000000000D000</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

The Web level permissions (0x0000000000000000) are specified as follows.

<table>
<thead>
<tr>
<th>Symbolic name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>0x00000000000001000</td>
<td>Allow access to the items contained within a site (2), list, or folder.</td>
</tr>
<tr>
<td>ViewPages</td>
<td>0x0000000000002000</td>
<td>Allow viewing of pages in a site (2).</td>
</tr>
<tr>
<td>AddAndCustomizePages</td>
<td>0x0000000000040000</td>
<td>Allow addition, modification, or deletion of HTML pages or Web Part pages, and editing of the site (2) using a compatible editor.</td>
</tr>
<tr>
<td>ApplyThemeAndBorder</td>
<td>0x0000000000080000</td>
<td>Allow application of a theme or borders to the entire site (2).</td>
</tr>
<tr>
<td>Symbolic name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ApplyStyleSheets</td>
<td>0x0000000000100000</td>
<td>Allow application of a style sheet (.css file) to the site (2).</td>
</tr>
<tr>
<td>ViewUsageData</td>
<td>0x0000000002000000</td>
<td>Allow viewing of reports on site usage.</td>
</tr>
<tr>
<td>CreateSSCSite</td>
<td>0x0000000004000000</td>
<td>Allow creation of a site (2) using Self-Service Site Creation, an implementation-specific capability.</td>
</tr>
<tr>
<td>ManageSubwebs</td>
<td>0x0000000008000000</td>
<td>Allow creation of a subsite within the site (2) or site collection.</td>
</tr>
<tr>
<td>CreateGroups</td>
<td>0x0000000001000000</td>
<td>Allow creation of a group (2) of users that can be used anywhere within the site collection.</td>
</tr>
<tr>
<td>ManagePermissions</td>
<td>0x0000000002000000</td>
<td>Allow creation and modification of permission levels on the site (2) and assigning permissions to users and site groups.</td>
</tr>
<tr>
<td>BrowseDirectories</td>
<td>0x0000000004000000</td>
<td>Allow enumeration of documents and folders in a site (2) using [MS-FPSE] and WebDAV interfaces.</td>
</tr>
<tr>
<td>BrowseUserInfo</td>
<td>0x0000000008000000</td>
<td>Allow viewing the information about all users of the site (2).</td>
</tr>
<tr>
<td>AddDelPrivateWebParts</td>
<td>0x0000000010000000</td>
<td>Allow addition or removal of personal Web Parts on a Web Part page.</td>
</tr>
<tr>
<td>UpdatePersonalWebParts</td>
<td>0x0000000020000000</td>
<td>Allow updating of Web Parts to display personalized information.</td>
</tr>
<tr>
<td>ManageWeb</td>
<td>0x0000000004000000</td>
<td>Allow all administration tasks for the site (2) as well as manage content.</td>
</tr>
<tr>
<td>AnonymousSearchAccessWebLists</td>
<td>0x0000000008000000</td>
<td>Allow content of lists and document libraries in the site (2) to be retrievable for anonymous users through SharePoint search if the list or document library has AnonymousSearchAccessList set.</td>
</tr>
<tr>
<td>UseClientIntegration</td>
<td>0x0000001000000000</td>
<td>Allow use of features that launch client applications; otherwise, users can only work on documents on their local machines and upload changes to the front-end Web server.</td>
</tr>
<tr>
<td>UseRemoteAPIs</td>
<td>0x0000002000000000</td>
<td>Allow use of SOAP, WebDAV, or [MS-FPSE] to access the site (2).</td>
</tr>
<tr>
<td>ManageAlerts</td>
<td>0x0000004000000000</td>
<td>Allow management of alerts for all users of the site (2).</td>
</tr>
<tr>
<td>CreateAlerts</td>
<td>0x0000008000000000</td>
<td>Allow creation of e-mail alerts.</td>
</tr>
<tr>
<td>EditMyUserInfo</td>
<td>0x0000010000000000</td>
<td>Allow users to change their own user information, such as adding a picture.</td>
</tr>
<tr>
<td></td>
<td>0x0000FE0000000000</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

The Special permissions (0xXXXX000000000000) are specified as follows.
<table>
<thead>
<tr>
<th>Symbolic name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0x3FFF000000000000</td>
<td>Reserved.</td>
</tr>
<tr>
<td>EnumeratePermissions</td>
<td>0x4000000000000000</td>
<td>Allow enumeration of permissions on the site (2), list, folder, document, or list item.</td>
</tr>
<tr>
<td></td>
<td>0x8000000000000000</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

### 2.2.3 Binary Structures

#### 2.2.3.1 Calendar View Options Type

Calendar View Options Type is a 1-byte value that specifies calendar options for front-end Web server display in the following format.

<table>
<thead>
<tr>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Sunday</td>
</tr>
<tr>
<td>001</td>
<td>Monday</td>
</tr>
<tr>
<td>010</td>
<td>Tuesday</td>
</tr>
<tr>
<td>011</td>
<td>Wednesday</td>
</tr>
<tr>
<td>100</td>
<td>Thursday</td>
</tr>
<tr>
<td>101</td>
<td>Friday</td>
</tr>
<tr>
<td>110</td>
<td>Saturday</td>
</tr>
</tbody>
</table>

**FirstDayOfWeek (3 bits):** An unsigned integer specifying the first day of the week. The following are valid values for the FirstDayOfWeek value.

<table>
<thead>
<tr>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Sunday</td>
</tr>
<tr>
<td>001</td>
<td>Monday</td>
</tr>
<tr>
<td>010</td>
<td>Tuesday</td>
</tr>
<tr>
<td>011</td>
<td>Wednesday</td>
</tr>
<tr>
<td>100</td>
<td>Thursday</td>
</tr>
<tr>
<td>101</td>
<td>Friday</td>
</tr>
<tr>
<td>110</td>
<td>Saturday</td>
</tr>
</tbody>
</table>

**FirstWeekOfYear (2 bits):** An unsigned integer specifying how the first week of the year is handled. The following are valid values for the FirstWeekOfYear value.

<table>
<thead>
<tr>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>The year starts on January 1.</td>
</tr>
<tr>
<td>01</td>
<td>The year starts with the first complete week.</td>
</tr>
<tr>
<td>10</td>
<td>The year starts with the first week of at least four days.</td>
</tr>
</tbody>
</table>

**WeekUI (1 bit):** If this bit is set, week numbers SHOULD be displayed in the front-end Web server.
Unused (2 bits): The last 2 bits of this structure are currently unused and MUST both be set to 0.

2.2.3.2 External Group Token

The External Group Token structure is a variable-length structure associated with a principal (2) that contains a collection of the systemIDs for the external groups of which the principal is a member.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| TimeTokenGenerated |
| ... |
| Size |
| Magic |
| AuthenticationType |
| UserSystemIdSize |
| TokenGroupsSize |
| Magic2 |
| UserSystemId (variable) |
| ... |
| TokenGroups (variable) |
| ... |

TimeTokenGenerated (8 bytes): An 8-byte, unsigned integer value specifying the time that this token was generated, stored as seconds since midnight, January 1, 1899, Coordinated Universal Time (UTC).

Size (4 bytes): A 4-byte, unsigned integer value specifying the length of the External Group Token in bytes.

Magic (4 bytes): A 4-byte, unsigned integer, which MUST use the value 0xCACBCECF.

AuthenticationType (4 bytes): A 4-byte, unsigned integer value specifying the authentication provider for the systemID of the principal. The value MUST be one of the following.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>Windows Integrated Authentication</td>
</tr>
<tr>
<td>0x00000003</td>
<td>ASP.NET Forms Authentication</td>
</tr>
</tbody>
</table>
**UserSystemIdSize (4 bytes):** A 4-byte, unsigned integer value specifying the length of the principal’s serialized binary `systemID`, in bytes.

**TokenGroupsSize (4 bytes):** A 4-byte, unsigned integer value specifying the length, in bytes, of the `TokenGroups` field containing the serialized binary systemIDs for the external groups.

**Magic2 (4 bytes):** A 4-byte, unsigned integer, which MUST use the value 0xDADBDEDF.

**UserSystemId (variable):** A variable-length field containing the serialized binary `systemID` for the principal, occupying the number of bytes specified in `UserSystemIdSize`.

**TokenGroups (variable):** A variable-length field containing a `Token Groups` structure, which contains the serialized binary `systemIDs` for the external groups of which the principal is a member, occupying the number of bytes specified in `TokenGroupsSize`.

### 2.2.3.3 Token Group Offset and Attributes

The `Token Group Offset and Attributes` structure specifies the length and offset of the serialized binary `systemID` for a corresponding external group within a `Token Groups` structure.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| Offset | Attributes |

**Offset (4 bytes):** A 4-byte, unsigned integer value specifying the offset in bytes from the beginning of the `Token Groups` structure to the beginning of the serialized binary `systemID` for this external group.

**Attributes (4 bytes):** A 4-byte, unsigned integer value specifying the length in bytes of the serialized binary `systemID` for this external group.

### 2.2.3.4 Token Groups

The `Token Groups` structure contains the serialized binary `SystemIDs` for a collection of external groups.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| GroupCount |
| OffsetsAndAttributes (variable) |
| ... |
| GroupSystemIds (variable) |
| ... |

**GroupCount (4 bytes):** A 4-byte, unsigned integer value specifying the number of external groups’ `SystemIDs` that are serialized in this structure.
OffsetsAndAttributes (variable): A variable-length serialized array containing GroupCount elements consisting of Token Group Offset and Attributes structures specifying the lengths and offsets of the serialized binary SystemIDs for the corresponding external groups, one for each external group's SystemID in the collection.

GroupSystemIds (variable): A variable-length field containing the collection of serialized binary SystemIDs for the external groups. Each external group's SystemID starts at the offset value in bytes from the beginning of the Token Groups structure and has the length in bytes specified by the Attributes value in the OffsetsAndAttributes array element corresponding to the external group.

2.2.3.5 WSS ACE
An ACE structure specifying the individual access rights of a principal (2).

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3 | 0 | 1 |
|   |   |   |   |   |   |   |   |   |   | PrincipalId |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   | PermMask |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   | ... |   |   |   |   |   |   |   |   |   |   |   |   |

PrincipalId (4 bytes): A 4-byte, signed integer specifying the User Identifier of the principal for this ACE.

PermMask (8 bytes): A WSS Rights Mask containing the list of rights that are granted to the principal.

2.2.3.6 WSS ACL Format
The WSS ACL Format structure contains an array of WSS_ACEs.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3 | 0 | 1 |
|   |   |   |   |   |   |   |   |   |   | Magic |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   | SecurityVersion |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   | ... |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   | NumAces |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   | Aces (variable) |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   | ... |   |   |   |   |   |   |   |   |   |   |   |   |

Magic (4 bytes): A 4-byte, unsigned integer describing the version of the ACL. This version of the protocol MUST use the value 0xfee3.

SecurityVersion (8 bytes): An 8-byte, signed integer specifying the site collection's security version value, which was used to compute the ACL. This value is not currently used.
**NumAces** (4 bytes): A 4-byte, unsigned integer specifying the count of **WSS_ACEs** within this ACL.

**Aces** (variable): An array of **WSS_ACEs** for each of the principals in this ACL.

### 2.2.3.7 WSS External Group Map Cache Format

The **WSS External Group Map Cache Format** structure contains a cache of WSS External Group Records mapping external groups to the site groups that contain them.

| 0 1 2 3 4 5 6 7 8 9 1 0 1 2 3 4 5 6 7 8 9 2 0 1 2 3 4 5 6 7 8 9 3 0 1 |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
|                              | **CachedVersion**           |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |
|                              | ...                        |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |
|                              | NumExternalGroupRecords    |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |
|                              | ExternalGroupRecords (variable) |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |
|                              | ...                        |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |

**CachedVersion** (8 bytes): An 8-byte, signed integer specifying the version of this cached information.

**NumExternalGroupRecords** (4 bytes): A 4-byte, signed integer specifying the count of **WSS External Group Records** present in this cache.

**ExternalGroupRecords** (variable): A serialized collection of **WSS External Group Records**.

### 2.2.3.8 WSS Compressed Structures

The **WSS Compressed structure** uses the ZLIB Compressed Data Format Specification version 3.3 to compress a binary or string value to a binary format when save it in to the database.

The header of the **WSS Compressed structure** is specified as follows.

| 0 1 2 3 4 5 6 7 8 9 1 0 1 2 3 4 5 6 7 8 9 2 0 1 2 3 4 5 6 7 8 9 3 0 1 |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
|                              | **ID**                      | **Version**                 |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |
|                              | **FileHeaderSize**           |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |
|                              | **OrigSize**                 |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |
|                              | Compressed Binary string (variable) |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |
|                              | ...                        |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |                              |

**ID** (2 bytes): MUST be 0xA8A9.

**Version** (2 bytes): MUST be 0x3031 (ASCII "01")
FileHeaderSize (4 bytes): A 4-byte, unsigned integer specifying the size of the header. The value is 0x0C000000.

OrigSize (4 bytes): A 4-byte, unsigned integer specifying the size of the original content. It MUST be the size of the uncompressed stream before compression.

Compressed Binary string (variable): The compressed string using zlib compression.

2.2.3.9 WSS External Group Record

The WSS External Group Record structure contains the mapping between an external group and the site groups that contain it.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| ExternalGroupSignatureSize |
| ExternalGroupSignature (variable) |
| ... |
| NumGroupIds |
| GroupIds (variable) |
| ... |

ExternalGroupSignatureSize (4 bytes): A 4-byte, signed integer specifying the size of the external group signature, in bytes.

ExternalGroupSignature (variable): An array of bytes specifying the signature for the external group using UTF-8 encoding. The number of bytes is specified by the ExternalGroupSignatureSize field.

NumGroupIds (4 bytes): A 4-byte, signed integer specifying the count of Site Group Identifiers in this record.

GroupIds (variable): An array of 4-byte, signed integers specifying the Site Group Identifiers that contain this external group. The number of elements is specified by the NumGroupIds field.

2.2.3.10 WSS User Token

The WSS User Token structure contains an array of Site Group Identifiers.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| Magic |
| TokenVersion |
| ... |
Magic (4 bytes): A 4-byte, unsigned integer specifying the version of the token format. This version of the protocol MUST use the value 0xdcd3.

TokenVersion (8 bytes): An 8-byte, signed integer specifying the site collection's security version value, which was used to compute the token. This value is not currently used and MUST be ignored.

NumGroupIds (4 bytes): A 4-byte, unsigned integer specifying the count of Site Group Identifiers within this token.

GroupIds (variable): An array of 4-byte integers for each of the site groups the corresponding user belongs to. The number of elements in the array is specified by the NumGroupIds field.

2.2.4 Result Sets

The following common result sets are used by this protocol.

2.2.4.1 ACL and Permission Result Set

The ACL and Permission Result Set contains information about the permissions associated with a security scope in effect for a document. The ACL and Permission Result Set is defined using T-SQL syntax, as follows.

```sql
Acl varbinary(max),
AnonymousPermMask bigint;
```

**Acl**: The binary serialization of the WSS ACL Format (section 2.2.3.6) access control list (ACL) for the security scope in effect.

**AnonymousPermMask**: A WSS Rights Mask (section 2.2.2.15) indicating the permissions granted to an anonymous user or a user who has no specific permissions on this document.

2.2.4.2 Custom Actions From Scope Result Set

This result set MUST return 1 row for each custom action retrieved. If there were no custom actions retrieved, this result set MUST NOT return any rows. This result set is defined using T-SQL syntax, as follows.

```sql
ScopeType int;
ScopeId uniqueidentifier;
Id uniqueidentifier;
Properties nvarchar(max);
Version nvarchar(64);
```

**ScopeType**: The custom action scope.

**ScopeId**: The site collection identifier, site identifier, or list identifier for which the custom action resides.
Id: The custom action identifier.

Properties: The custom action data describing its functionality.

Version: The custom action version.

2.2.4.3 Domain Group Cache BEDS Update Result Set

The Domain Group Cache BEDS Update Result Set contains information to be used in recomputing the domain group map cache, which contains the mapping of external groups to the site groups that they are members of. The presence of the Domain Group Cache BEDS Update Result Set means the database's copy of the domain group map cache is out of date and MUST be recomputed to ensure that proper security checks can be made.

The Domain Group Cache BEDS Update Result Set is defined using T-SQL syntax, as follows.

```sql
CREATE TABLE DomainGroupCacheBEDSUpdateResultSet (tp_id int, tp_SystemId varbinary(512), GroupId int);
```

tp_id: The identifier of an external group which is a member of a Site Group.

tp_SystemId: The SystemId of the external group.

GroupId: The Site Group Identifier of the Site Group containing the given domain group.

2.2.4.4 Domain Group Cache Versions Result Set

The Domain Group Cache Versions Result Set contains information about the version numbers associated with the Domain Group Map Caches on the front-end Web server and on the back-end database server for the specified Site Collection. A Domain Group Map Cache contains a serialized representation of the External Groups that are members of Site Groups in the Site Collection. The version numbers in this result set can be used to determine whether the Domain Group Map Caches on the front-end Web server or on the back-end database server are out of date and need to be refreshed. A special version number value of -2 indicates that the value MUST NOT be used for comparison.

When the Domain Group Cache Versions Result Set is returned, it MUST contain one row of version number data.

The Domain Group Cache Versions Result Set is defined using T-SQL syntax, as follows.

```sql
CREATE TABLE DomainGroupCacheVersionsResultSet (RealVersion bigint, CachedVersion bigint, FrontEndVersion bigint);
```

RealVersion: The most recent version number of the domain group Map Cache information.

CachedVersion: The version number of the domain group Map Cache information on the back-end database server.

FrontEndVersion: The version number of the domain group Map Cache information on the front-end Web server.
2.2.4.5 Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the
domain group map cache. If the Domain Group Cache WFE Update Result Set is returned, it
indicates that the Back-end database server domain group map cache is up-to-date, and the front-end
Web server cache can be refreshed if necessary.

The Domain Group Cache WFE Update Result Set is defined using T-SQL syntax, as follows.

```
DomainGroupMapCache varbinary(max);
```

**DomainGroupMapCache**: A column containing the serialized domain group map cache data. If the
value of **FrontEndVersion** is greater than or equal to the value of **CachedVersion** in the results in
the Domain Group Cache Versions Result Set (section 2.2.4.4), this MUST be NULL.

2.2.4.6 Document Content Metadata Result Set

The Document Content Metadata Result Set contains the metadata for the document’s binary
stream.

```
{Size} int,
{SiteRbsCollectionId} int,
{Version} int,
{InternalVersion} int,
{HistVersion} int,
{Id} uniqueidentifier,
{DirName} nvarchar(256),
{LeafName} nvarchar(128),
{ParentId} uniqueidentifier,
{SetupPathVersion} tinyint,
{SetupPath} nvarchar(255),
{Dirty} bit,
{DocFlags} int,
{Level} tinyint,
{DoclibRowId} int,
{VirusVendorID} int,
{VirusStatus} int,
{VirusInfo} nvarchar(255),
{VirusInfoEx} varbinary(max),
{ContentVersion} int,
{NextBSN} bigint,
{StreamSchema} byte,
{SiteId} uniqueidentifier
```

**{Size}**: The size of the document stream, in bytes.

**{SiteRbsCollectionId}**: The identifier for the remote BLOB storage collection for the site
collection, or zero if remote BLOB storage is not configured for this database.

**{Version}**: The current **displayed version** number of the document.

**{InternalVersion}**: The current **internal version number** of the document.

**{HistVersion}**: The user interface (UI) version number for the requested document.

**{Id}**: The **document identifier** of the requested document.

**{DirName}**: The **directory name** of the requested document.

**{LeafName}**: The **leaf name** of the requested document.
{ParentId}: The document identifier of the item's parent container.

{SetupPathVersion}: For an uncustomized document, this parameter governs the setup path location to which the {SetupPath} fragment is relative. This value MUST be NULL if the document does not exist, and it is undefined for a document that was never uncustomized. The value MUST be one of the following.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Relative to the install location of Windows SharePoint Services 2.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).</td>
</tr>
<tr>
<td>3</td>
<td>Relative to the install location of Windows SharePoint Services 3.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).</td>
</tr>
<tr>
<td>4</td>
<td>Relative to the install location of Microsoft SharePoint Foundation 2010 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).</td>
</tr>
<tr>
<td>15</td>
<td>Relative to the install location of Microsoft SharePoint Foundation 2013 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\15).</td>
</tr>
<tr>
<td>16</td>
<td>Relative to the install location of Microsoft SharePoint Server 2016 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\16).</td>
</tr>
</tbody>
</table>

{SetupPath}: For a document that has never been uncustomized, this contains the setup path fragment relative to the base setup path where the content stream of this document can be found, as described in {SetupPathVersion}. Otherwise, this parameter MUST be NULL.

{Dirty}: This parameter MUST be set to one, if this document has had dependency update processing performed; otherwise, it MUST be set to zero. If the document does not have a content stream, the value is implementation-dependent and MUST be ignored.

{DocFlags}: A Doc Flags value (section \ref{DocFlags}) describing the document.

{Level}: A publishing level value specifying the publishing status of this document.

{DoclibRowId}: The identifier for a row in a document library for this document. If the requested document is not contained in a list, this value MUST be NULL.

{VirusVendorID}: The identifier of the anti-virus vendor that processed this document. This value MUST be NULL, if the document has not been processed by an anti-virus scanner.

{VirusStatus}: The Virus Status (section \ref{VirusStatus}) of the document. This value MUST be NULL, if the requested document does not exist or if it has not been processed by an anti-virus scanner.

{VirusInfo}: An anti-virus scanner specific message returned by the anti-virus scanner when it last processed the document.

{VirusInfoEx}: A binary value that specifies extensible virus information for this document, which is opaque to the back-end server.

{ContentVersion}: A counter used for internal conflict detection that is incremented any time a change is made to the binary contents of this document.

{NextBSN}: The current binary large object (BLOB) sequence number of the requested document.

{StreamSchema}: The current stream schema of the document being requested.

{SiteId}: The site collection identifier of the site collection containing the document being requested.
### 2.2.4.7 Document Content Stream Result Set

The **Document Content Stream Result Set** contains a row for each of the document's stream binary pieces.

- **{ExpirationUTC}**: The time stamp, in UTC format, of when this stream binary piece expires.
- **{DocId}**: The document identifier of the requested document.
- **{SiteId}**: The site collection identifier of the site collection containing the requested document.
- **{Partition}**: The stream partition that this stream binary piece belongs to.
- **{BSN}**: The BLOB sequence number of this stream binary piece.
- **{StreamId}**: The stream identifier of this stream binary piece.
- **{Type}**: The stream type of this stream binary piece.
- **{Size}**: The size of the stream binary piece, in bytes.
- **{Content}**: The data of the stream binary piece.
- **{RbsResReference}**: This value is used in remote BLOB storage and is opaque to the back-end database server. For additional information regarding remote BLOB storage, see [MS-WSSO] section 2.1.2.3.8.

### 2.2.4.8 Document Metadata Result Set

The **Document Metadata Result Set** returns the metadata for a document.

The **Document Metadata Result Set** is defined using T-SQL syntax, as follows.

```
Id, uniqueidentifier,
{FullUrl}, nvarchar(260),
Type, tinyint,
MetaInfoTimeLastModified, datetime,
MetaInfo, varbinary(max),
Size, int,
TimeCreated, datetime,
TimeLastModified, datetime,
ETagVersion, int,
DocFlags, int,
{ListType}, int,
{tp_Name}, int,
{ListTitle}, int,
{CacheParseId}, uniqueidentifier,
{GhostDirName}, int,
{GhostLeafName}, int,
tp_Login, nvarchar(255),
```
CheckoutDate, CheckoutExpires, VirusStatus, VirusInfo, VirusInfoEx, SetupPathVersion, SetupPath, SetupPathUser, NextToLastTimeModified, UIVersion, CheckinComment, WelcomePageUrl, WelcomePageParameters, {tp_Flags}, Acl, AnonymousPermMask, DraftOwnerId, Level, ParentVersion, TransformerId, ParentLeafName, ProgId, DoclibRowId, {tp_DefaultWorkflowId}, ListId, ItemChildCount, FolderChildCount, MetaInfoVersion, {CurrentVerMetaInfo}, ContentVersion, {ContentVersionDirty}, {NextBSN}, {StreamSchema}, {InternalVersion}

**Id:** The document identifier of the requested document.

**{FullUrl}:** The full store-relative form URL for the document being requested.

**Type:** The Document Store Type (section 2.2.2.4) of this document.

**MetaInfoTimeLastModified:** A time stamp in UTC format specifying the last time the MetaInfo value of this document was changed, which is useful for providing minimal metadata returns to clients. This value MUST be NULL, if the MetaInfo column value of the document has never been changed.

**MetaInfo:** A metadict for this document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value MUST be NULL if the document does not have any metadict associated with it.

**Size:** The number of bytes in the document stream of the document. This value can be NULL if the document is a folder or site Document Store Type.

**TimeCreated:** A time stamp in UTC format that specifies when this document was created.

**TimeLastModified:** A time stamp in UTC format that specifies when the document was last saved. This value does not necessarily correspond to the actual time when the document was last modified.

**ETagVersion:** A counter used for internal conflict detection that is incremented any time a change is made to this document.

**DocFlags:** A Doc Flags value (section 2.2.2.3) describing the document. This value can be NULL.

**{ListType}:** This value MUST be NULL.

**{tp_Name}:** This value MUST be NULL.
{ListTitle}: This value MUST be NULL.

{CacheParseId}: This value MUST be NULL.

{GhostDirName}: A placeholder for a directory name. This value MUST be NULL.

{GhostLeafName}: A placeholder for a leaf name. This value MUST be NULL.

tp_Login: If this document exists and is currently checked out, this is the login name of the user to whom it is checked out. In all other cases, this is NULL.

CheckoutDate: A time stamp in UTC format specifying when this document was checked out. This value MUST be NULL if the document has never been checked out.

CheckoutExpires: A time stamp in UTC format that specifies when the short-term lock for this document will expire. If this date is in the past, the document SHOULD be treated as unlocked. This value can be NULL if no short-term lock has been placed on the document.

VirusStatus: An enumerated type specifying the current virus state of this document. This value can be NULL if it has not been processed by a virus scanner. Valid values are listed in the Virus Status (section 2.2.1.2.17).

VirusInfo: A string containing a provider specific message returned by the virus scanner when it last processed the document. This value can be NULL if it has not been processed by a virus scanner.

VirusInfoEx: A binary value that specifies extensible virus information for this document, which is opaque to the back-end server.

SetupPathVersion: For an uncustomized document, this governs the setup path location with which the {SetupPath} fragment is relative. This value MUST NOT be NULL. The following values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).</td>
</tr>
<tr>
<td>3</td>
<td>The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).</td>
</tr>
<tr>
<td>4</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Foundation 2010 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).</td>
</tr>
<tr>
<td>15</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Foundation 2013 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\15).</td>
</tr>
<tr>
<td>16</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Server 2016 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\16).</td>
</tr>
</tbody>
</table>

SetupPath: For a document that is now or once was uncustomized, this contains the setup path fragment relative to the base setup path described earlier by the SetupPathVersion value, where the content stream of this document can be found. This value can be NULL.

SetupPathUser: If this document is now or once was uncustomized, this contains the login name of the user that created the uncustomized document. This value can be NULL.

NextToLastTimeModified: The value of TimeLastModified from the previous time when the document was last saved. If the NextToLastTimeModified value matches the TimeLastModified value when the rename occurred and the client has a document that it has successfully fixed up, the
client can safely submit the document to the front-end Web server despite what appears to be an intervening edit to the document. This value can be NULL.

**UIVersion:** The UI version number for the document. This value MUST be NULL in the case of a document that does not exist.

**CheckinComment:** An optional description provided when a document is checked in or published, which could be displayed in the version management UI. This value can be NULL.

**WelcomePageUrl:** If this document is a folder, this specifies an optional page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself and MUST be subsumed by that folder. Attempts to break out of the folder such as "../../somepage.aspx" are not valid. This value can be NULL.

**WelcomePageParameters:** Contains optional URL parameters to specify as part of the WelcomePageUrl value. These parameters start at either the query string signifier '?' or the bookmark signifier '#'. This value can be NULL.

**{tp_FLAGS}:** A List Flags value (section 2.2.2.5) describing the list that contains this document.

**Acl:** The binary serialization of the WSS ACL Format (section 2.2.3.6) ACL for this document. This is either explicitly defined or inherited from the parent object of the document. This value can be NULL if a WSS ACL is not defined for the document.

**AnonymousPermMask:** A WSS Rights Mask (section 2.2.2.15) that indicates the permissions granted to a user that is anonymous, or has no specific permissions, to this document. This value can be NULL if anonymous access to the document is not allowed.

**DraftOwnerId:** The User Identifier (section 2.2.1.1.13) of the user that published this document as a draft. This value MUST only be non-NULL if the requested document exists and is a draft version.

**Level:** A Publishing Level Type value (section 2.2.2.6) specifying the publishing status of this document.

**ParentVersion:** If the document is a transformed version of another document, this is the user interface (UI) version value from the original document. This value MUST be NULL if the document is not the product of a document transformation.

**TransformerId:** If the document is a transformed version of another document, this is the GUID of the agent that performed the transformation. This value MUST be NULL if the document is not the product of a document transformation.

**ParentLeafName:** If the document is a transformed version of another document and the original document is in the same folder as the transformed document, this is the leaf name of the original document. If either is moved, the relationship is broken. This value MUST be NULL if the document is not the product of a document transformation.

**ProgId:** Designates a preferred application to open the document. The ProgId is used to distinguish between different applications that save files with a given file extension (that is, different editors for HTML or XML files). This value MUST be NULL if the parser did not specify a ProgId when the document was saved.

**DoclibRowId:** The identifier for a row in a document library for this document. If the requested document is not contained in a list, this value MUST be NULL.

**{tp_DefaultWorkflowId}:** The Workflow Identifier (section 2.2.1.1.16) corresponding to the workflow to be invoked if this document is in a moderated list and is submitted for approval as part of a check in. This value MUST be NULL.

**ListId:** The List Identifier (section 2.2.1.1.5) of the list that contains the requested document. If the document is not contained in a list, this value MUST be NULL.
**ItemChildCount**: The number of non-folder (those whose **Document Store Type** is not folder) children of this document.

**FolderChildCount**: The number of folder (those whose **Document Store Type** is folder) children of this document.

**MetaInfoVersion**: A counter used for internal conflict detection that is incremented any time a change is made to the metadict for this document.

**{CurrentVerMetaInfo}**: This value is not used and MUST be NULL.

**ContentVersion**: A counter used for internal conflict detection that is incremented any time a change is made to the binary contents of this document. This value MUST be NULL if the document does not exist.

**{ContentVersionIsDirty}**: This value specifies whether the **ContentVersion** value can be used for conflict detection. If the **ContentVersion** value can be used for conflict detection or the **ContentVersion** value is NULL, this value MUST be zero; otherwise, the value MUST be one.

**{NextBSN}**: The current BLOB sequence number of the document being returned.

**{StreamSchema}**: The current stream schema of the document being returned.

**{InternalVersion}**: The internal version number of the document being returned.

### 2.2.4.9 Document Version Metadata Result Set

The **Document Version Metadata Result Set** contains the metadata about the requested version of the document.

The **Document Version Metadata Result Set** MUST be returned only if **@Version** is not negative. If the document exists, the **Document Version Metadata Result Set** MUST contain one row; otherwise, it MUST contain no rows.

The **Document Version Metadata Result Set** is defined using T-SQL syntax, as follows.

```sql
Id                             uniqueidentifier,
{FullUrl}                      nvarchar(260),
Type                           tinyint,
MetaInfoTimeLastModified       datetime,
MetaInfo                       varbinary(max),
Size                           int,
TimeCreated                    datetime,
ETagVersion                    int,
DocFlags                       int,
{ListType}                     int,
{tp_Name}                      nvarchar(38),
{ListTitle}                    nvarchar(255),
{CacheParseId}                 uniqueidentifier,
{GhostDirName}                 nvarchar(256),
{GhostLeafName}                nvarchar(128),
tp_Login                       nvarchar(255),
CheckoutDate                   datetime,
CheckoutExpires                datetime,
VirusStatus                    int,
VirusInfo                      nvarchar(255),
VirusInfoEx                    varbinary(max),
SetupPathVersion               tinyint,
SetupPath                      nvarchar(255),
SetupPathUser                  nvarchar(255),
NextToLastTimeModified         datetime,
UIVersion                      int,
```
Id: The document identifier of the requested document.

{FullUrl}: The full store-relative form URL for the document being requested.

Type: The Document Store Type (section 2.2.2.4) of this document.

MetaInfoTimeLastModified: A time stamp in UTC format specifying the last time the Metainfo value of this document was changed. This value MUST be NULL if the Metainfo column value of the document has never been changed.

MetaInfo: A metadict for this document version. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value MUST be NULL if the document does not have any metadict associated with it.

Size: The number of bytes in the document stream of the document version requested. This value can be NULL if the document is a folder or site Document Store Type (section 2.2.2.4).

TimeCreated: A time stamp in UTC format specifying when this document was created.

TimeCreated: A time stamp in UTC format specifying when the document was last saved. This corresponds to the TimeCreated or TimeLastModified of the document version requested.

ETagVersion: A counter used for internal conflict detection that is incremented any time a change is made to this document.

DocFlags: A Doc Flags (section 2.2.2.3) value describing this document. This value can be NULL if the document is a folder Document Store Type.

{ListType}: This value MUST be NULL.

{tp_Name}: This value MUST be NULL.

{ListTitle}: This value MUST be NULL.

{CacheParseId}: This value MUST be NULL.

{GhostDirName}: A placeholder for a directory name. This value MUST be NULL.
{GhostLeafName}: A placeholder for a leaf name. This value MUST be NULL.

tp_Login: If this document exists and is currently checked out, this is the login name of the user to whom it is checked out. In all other cases, this is NULL.

CheckoutDate: A time stamp in UTC format specifying when this document was checked out. This value can be NULL if the document is checked out.

CheckoutExpires: A time stamp in UTC format specifying when the short-term lock for this document will expire. If this date is in the past, the document SHOULD be treated as unlocked. This value can be NULL if no short-term lock has been placed on the document.

VirusStatus: An enumerated type specifying the current virus state of this document version. This value can be NULL if it has not been processed by a virus scanner. Valid values are listed in the Virus Status section 2.2.1.2.17.

VirusInfo: A string containing a provider specific message returned by the virus scanner when it last processed the document. This value can be NULL if it has not been processed by a virus scanner.

VirusInfoEx: A binary value that specifies extensible virus information for this document, which is opaque to the back-end server.

SetupPathVersion: For an uncustomized document, this governs the setup path location that the SetupPath fragment is relative to. This value MUST NOT be NULL. The following are values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).</td>
</tr>
<tr>
<td>3</td>
<td>The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).</td>
</tr>
<tr>
<td>4</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Foundation 2010 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).</td>
</tr>
<tr>
<td>15</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Foundation 2013 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\15).</td>
</tr>
<tr>
<td>16</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Server 2016 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\16).</td>
</tr>
</tbody>
</table>

SetupPath: For a document that is now or once was uncustomized, this contains the setup path fragment relative to the base setup path described earlier by the SetupPathVersion value, where the content stream of this document can be found. This value can be NULL.

SetupPathUser: If this document is now or once was uncustomized, this contains the login name of the user that created the uncustomized document. This value can be NULL.

NextToLastTimeModified: The value of TimeLastModified from the previous time when the document was last saved. If the NextToLastTimeModified value matches the TimeLastModified value when the rename occurred and the client has a document that it has successfully fixed up, the client can safely submit the document to the front-end Web server despite what appears to be an intervening edit to the document. This value can be NULL.
**UIVersion:** The user interface (UI) version number for the document. This value can be produced as the integer counter described earlier (the first Version field). This value MUST match @Version.

**CheckinComment:** An optional description provided when a document is checked in, which could be displayed in version management UI. This value can be NULL.

**WelcomePageUrl:** If this document is a folder, this specifies an optional page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself and MUST be subsumed by that folder. Attempts to break out of the folder such as "../somepage.aspx" are not valid. This value can be NULL.

**WelcomePageParameters:** Contains optional URL parameters to specify as part of the WelcomePageUrl value. These parameters start at either the query string signifier '?' or the bookmark signifier '#'. This value can be NULL.

**(tp_Flags):** A List Flags value (section 2.2.5) describing the list that contains this document. This value MUST be NULL.

**Acl:** The binary serialization of the WSS ACL Format (section 2.2.3.6) ACL for this document. This is either explicitly defined or inherited from the parent object of the document.

**AnonymousPermMask:** A WSS Rights Mask (section 2.2.15) that indicates the permissions granted to a user that is anonymous, or has no specific permissions, to this document. This value can be NULL if anonymous access to the document is not allowed.

**DraftOwnerId:** The User Identifier (section 2.2.1.1.13) of the user that published this document as a draft. This value is only non-NULL if the requested document is a draft version.

**Level:** A Publishing Level Type (section 2.2.6) value specifying the publishing status of this document. This value MUST match @Level.

**ParentVersion:** If the document is a transformed version of another document (see Doc Flags value 0x00000400), this is the UIVersion value from the parent document. This value MUST be NULL if the document is not a child of a document transformation.

**TransformerId:** If the document is a transformed version of another document, this is the GUID of the agent that performed the transformation. This value MUST be NULL if the document is not the product of a document transformation.

**ParentLeafName:** If the document is a transformed version of another document and the original document is in the same folder as the transformed document, this is the leaf name of the original document. If either is moved, the relationship is broken. This value MUST be NULL if the document is not the product of a document transformation.

**ProgId:** Designates a preferred application to open the document. The ProgId is used to distinguish between different applications that save files with a given file extension (that is, different editors for HTML or XML files). This value MUST be NULL if the parser did not specify a ProgId when the document was saved.

**DoclibRowId:** The identifier for a row in a document library for this document. If the requested document is not contained in a list, this value MUST be NULL.

**(tp_DefaultWorkflowId):** A placeholder for the Workflow Identifier (section 2.2.1.16) corresponding to a workflow invoked as part of a check in on a moderated list. This value MUST be NULL.

**ListId:** The List Identifier (section 2.2.1.5) of the list containing the requested document. If the document is not contained in a list, this value MUST be NULL.

**ItemChildCount:** The number of non-folder (those whose Document Store Type is not folder) children of this document.
FolderChildCount: The number of folder (those whose Document Store Type is folder) children of this document.

MetaInfoVersion: A counter incremented any time a change is made to the metainfo for this document and used for internal conflict detection.

{MetaInfo}: A metadict for this document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value MUST be NULL if the document does not have any metadict associated with it.

ContentVersion: A counter used for internal conflict detection that is incremented any time a change is made to the binary contents of this document. This value MUST be NULL if the document does not exist.

{ContentVersionIsDirty}: This value specifies whether the ContentVersion value can be used for conflict detection. If the ContentVersion value can be used for conflict detection or the ContentVersion value is NULL, this value MUST be zero; otherwise, it MUST be one.

{NextBSN}: The current BLOB sequence number of the document being returned.

{StreamSchema}: The current stream schema of the document being returned.

{InternalVersion}: The internal version number of the document being returned.

2.2.4.10 Empty Result Set

The Empty Result Set is defined using T-SQL syntax, as follows.

{No column name} Not Applicable

2.2.4.11 Event Receivers Result Set

The Event Receivers Result Set contains information about the event receivers defined for an event host. This result set MUST contain one row for each event receiver registered for the event host.

The Event Receivers Result Set is defined using T-SQL syntax, as follows.

Id uniqueidentifier,
Name nvarchar(256),
SiteId uniqueidentifier,
WebId uniqueidentifier,
HostId uniqueidentifier,
HostType int,
ItemId int,
DirName nvarchar(256),
LeafName nvarchar(128),
Synchronization int,
Type int,
SequenceNumber int,
RemoteUrl nvarchar(4000),
Assembly nvarchar(256),
Class nvarchar(256),
SolutionId varbinary(512),
Data nvarchar(256),
Filter nvarchar(256),
SourceId varbinary(512),
SourceType int,
Credential int,
ContextType varbinary(16),
ContextEventType varbinary(16),
ContextId varbinary(16),
**ContextObjectId**: varbinary(16),
**ContextCollectionId**: varbinary(16),
**Hash**: nvarchar(50),
**ValidatorsHash**: char(64),
**ValidationErrorUrl**: nvarchar(4000),
**ValidationErrorMessage**: nvarchar(4000)

**Id**: The **Event Receiver Identifier** (section 2.2.1.3) of the event receiver.

**Name**: The **display name** of the event receiver.

**SiteId**: The **site collection identifier** for a **site collection** that contains the event host on which the event receiver is registered.

**WebId**: The **site identifier** for the **site (2)** that contains the event host on which the event receiver is registered.

**HostId**: The **document identifier** for the event host on which the event receiver is registered.

**HostType**: The **Event Host Type** (section 2.2.1.2.5) of the event host on which the event receiver is registered.

**ItemId**: Reserved. The server MUST default this value to NULL and a client MUST NOT use this value.

**DirName**: Reserved. The server MUST default this value to NULL and a client MUST NOT use this value.

**LeafName**: Reserved. The server MUST default this value to NULL and a client MUST NOT use this value.

**Synchronization**: Contains the execution behavior of the event receiver. The value MUST be one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>If the <strong>Event Receiver Type</strong> (section 2.2.1.6) specified by the <strong>Type</strong> column is less than 10001, the execution behavior of the event receiver MUST be synchronous.</td>
</tr>
<tr>
<td>1</td>
<td>The execution behavior of the event receiver MUST be synchronous.</td>
</tr>
<tr>
<td>2</td>
<td>The execution behavior of the event receiver MUST be asynchronous.</td>
</tr>
</tbody>
</table>

**Type**: The **Event Receiver Type** of the event receiver.

**SequenceNumber**: An ordinal value that determines the relative order in which the event receiver is triggered. **SequenceNumber** MUST be greater than or equal to zero, and less than or equal to 65535.

**RemoteUrl**: The **URL** of the remote event receiver.

**Assembly**: The name of the .NET assembly that contains the implementation of the event receiver.

**Class**: The name of the .NET class definition that contains the implementation of the event receiver.

**SolutionId**: The sandboxed solution identifier of the **sandboxed solution**.

**Data**: Additional data persisted on behalf of the event receiver implementation, to be passed to the event receiver.

**Filter**: Reserved. The server MUST default this value to NULL and a client MUST NOT use this value.
**SourceId:** The feature identifier or content type identifier of the feature or content type that registered the event receiver. If the event receiver was not registered by a feature or content type, the value MUST be NULL.

**SourceType:** An integer value that specifies the source of the registration for the event receiver. The value MUST be one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The event receiver does not come from a specially treated source.</td>
</tr>
<tr>
<td>1</td>
<td>A content type registered the event receiver.</td>
</tr>
<tr>
<td>2</td>
<td>A feature registered the event receiver.</td>
</tr>
</tbody>
</table>

**Credential:** Reserved. The server MUST default this value to NULL and a client MUST NOT use this value.

**ContextType:** The workflow identifier of the workflow that the event (2) relates to, if any.

**ContextEventType:** Reserved. The server MUST default this value to NULL and a client MUST NOT use this value.

**ContextId:** The workflow identifier for the registered workflow, if any.

**ContextObjectId:** The document identifier for the object instance that the workflow is registered against, if any.

**ContextCollectionId:** A workflow identifier for the workflow instance that manages the event receiver, if any.

**Hash:** The implementation-specific hash of the content of the sandboxed solution.

**ValidatorsHash:** The implementation-specific hash of the sandboxed solution validator that validated the sandboxed solution.

**ValidationErrorUrl:** The URL to render the sandboxed solution validator validation failed. If validation succeeded, it MUST be NULL.

**ValidationErrorMessage:** If the sandboxed solution validator validation failed, this stores an error message string. Otherwise, it MUST be NULL.

### 2.2.4.12 Individual URL Security Result Set

The Individual URL Security Result Set contains security information about a document.

The Individual URL Security Result Set is defined using T-SQL syntax, as follows.

```sql
{ListId} uniqueidentifier,
Acl varbinary(max),
AnonymousPermMask bigint,
{IsAttachment} bit,
{NeedManageListRight} bit,
{BaseType} int,
{ExcludedType} int,
{ListFlags} tinyint,
{DraftOwnerId} int,
{DocType} tinyint,
{bNeedNoThrottle} bit,
{ItemCount} int,
{DoclibRowId} int,
```
{ListID}: The List Identifier of the list or document library containing the document location.

Acl: The WSS ACL for the security scope associated with the document location.

AnonymousPermMask: The WSS Rights Mask that applies to an anonymous user, or a user with no assigned permissions, in the security scope associated with the document location.

{IsAttachment}: A bit flag specifying whether the document location is an attachment or an attachment folder. This value MUST be set to 1 if the document location is an attachment or attachment folder; otherwise, it MUST be 0.

{NeedManageListRight}: A bit flag specifying whether the current user needs the ManageLists bit of the WSS Rights Mask set to write to the document location. This value MUST be set to 1 if the current user MUST have the ManageLists bit of the WSS Rights Mask set in the security scope of the document location to write the document; otherwise, it MUST be 0.

{BaseType}: The List Base Type of the list or document library containing the document location.

{ExcludedType}: Contains an enumerated value specifying whether the document location is within a special folder type in the containing list or document library. The following values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The document location is not contained in a special folder.</td>
</tr>
<tr>
<td>1</td>
<td>The document location is, or is contained within, a forms folder: a folder named &quot;Forms&quot; within a document library.</td>
</tr>
<tr>
<td>2</td>
<td>The document location is, or is contained within, a Web image folder: a folder named &quot;.w&quot; within a document library.</td>
</tr>
<tr>
<td>3</td>
<td>The document location is, or is contained within, a thumbnail folder: a folder named &quot;.t&quot; within a document library.</td>
</tr>
<tr>
<td>4</td>
<td>The document location is at the root folder of the list or document library.</td>
</tr>
</tbody>
</table>

{ListFlags}: The List Flags of the list or document library containing the document location.

{Level}: The Publishing Level Type value specified for the document, or if the parameter was NULL, the Publishing Level Type of the current version of the document for the current user.

{DraftOwnerId}: The User Identifier of the principal (2) that published this document as a draft. This value MUST be NULL if the document does not exist or does not have a draft version.

{DocType}: The Document Store Type (section 2.2.2.4) of this document.

{bNeedNoThrottle}: If the list or document library containing the document location is a list or document library that can ignore throttle, then this value MUST be 1. Otherwise, it MUST be 0.

{ItemCount}: The number of the Items in list or document library containing the document location is a list or document library.

{DoclibRowId}: The row identifier of the document in the containing list or document library.

DocFlags: A Doc Flags (section 2.2.2.3) value specifying information about the document.
2.2.4.13  Link Information Result Set

The Link Information Result Set returns information about each forward link from the document and backward link to the document within the site collection.

The Link Information Result Set is defined using T-SQL syntax, as follows.

```
LinkDirName          nvarchar(256),
LinkLeafName         nvarchar(128),
LinkType             tinyint,
LinkSecurity         tinyint,
LinkDynamic          tinyint,
LinkServerRel        bit,
LinkStatus           tinyint,
PointsToDir          bit,
WebPartId            uniqueidentifier,
LinkNumber           int,
WebId                uniqueidentifier,
Search               nvarchar(max),
FieldId              uniqueidentifier;
```

**LinkDirName:** Contains the directory name of the link target.

**LinkLeafName:** Contains the leaf name of the link target.

**LinkType:** The LinkType value of the link. This value MUST be NULL for a backward link.

**LinkSecurity:** The LinkSecurity value of the forward link, which specifies whether the scheme of the link is HTTP or HTTPS. This value MUST be NULL for a backward link.

**LinkDynamic:** The LinkDynamic value of the forward link, which specifies whether the link is to a special type of target. This value MUST be NULL for a backward link.

**LinkServerRel:** A bit flag which specifies whether the link URL is server-relative URL or not. A value of 1 specifies a server-relative URL. This value MUST be NULL for a backward link.

**LinkStatus:** The Document Store Type value of the document targeted by a forward link. This value MUST be 128 for a backward link. If the forward link target is a document that does not exist, or if this link refers to a target that exists outside the specified site collection, or if it refers to a location that could not be verified, this value MUST be NULL.

**PointsToDir:** A bit flag specifying whether the target of the forward link was a directory and has been modified to target a welcome page. This value MUST be NULL for a backward link. For a forward link, if the link target is a directory where a welcome page is specified, the link MUST be changed to the URL of the welcome page and PointsToDir MUST be set to 1 so that the link can be distinguished from an explicit link to the welcome page; otherwise this value MUST be 0.

**WebPartId:** This value MUST be NULL.

**LinkNumber:** This value MUST be NULL.

**WebId:** This value MUST be NULL.

**Search:** This value MUST be NULL.

**FieldId:** This value MUST be NULL.

2.2.4.14  List Metadata Result Set

The List Metadata Result Set contains the metadata associated with a list.
The **List Metadata Result Set** is defined using T-SQL syntax, as follows.

```sql
tp_Id                                uniqueidentifier,
tp_Title                             nvarchar(255),
tp_Modified                          datetime,
tp_Created                           datetime,
tp_LastDeleted                       datetime,
tp_Version                           int,
tp_BaseType                          int,
tp_FeatureId                         uniqueidentifier,
tp_ServerTemplate                   int,
DirName                              nvarchar(256),
LeafName                             nvarchar(128),
DirName                              nvarchar(256),
LeafName                             nvarchar(128),
tp_ReadSecurity                      int,
.tp_WriteSecurity                    int,
.tp_Description                      nvarchar(max),
{tp_Fields}                          varbinary(max),
.tp_Direction                        int,
AnonymousPermMask                    bigint,
.tp_Flags                            bigint,
.tp_ThumbnailSize                    int,
.tp_ThumbnailHeight                  int,
.tp_ThumbnailWidth                    int,
.tp_ItemCount                        int,
.tp_Author                           int,
.tp_HasInternalFGP                   bit,
.tp_ScopeId                          uniqueidentifier,
Acl                                  varbinary(max),
tp_EventSinkAssembly                 nvarchar(255),
tp_EventSinkClass                    nvarchar(255),
tp_EventSinkData                     nvarchar(255),
tp_EmailAlias                        nvarchar(128),
tp_WebFullUrl                        nvarchar(256),
tp_WebId                             uniqueidentifier,
tp_WebTitle                          nvarchar(255),
tp_Web                               int,
tp_WebLanguage                      int,
tp_WebCollation                      smallint,
tp_SendToLocation                   nvarchar(512),
{tp_MaxMajorVersionCount}            int,
{tp_MaxMajorWithMinorVersionCount}   int,
.tp_DefaultWorkflowId                uniqueidentifier,
{HasBackLookupRels}                 bit,
tp_ContentTypes                      varbinary(max),
.tp_Subscribed                      bit,
{ChildCount}                         int,
tp_NoThrottleListOperations         bit,
tp_TitleResource                     nvarchar(256),
tp_DescriptionResource               nvarchar(256),
tp_ValidationFormula                 nvarchar(1024),
tp_ValidationMessage                 nvarchar(1024),
Followable                           bit;
```

**tp_Id**: The List Identifier of the list.

**tp_Title**: The title of this list for display in the front-end Web server.

**tp_Modified**: A time stamp in UTC format specifying when this list was last modified.

**tp_Created**: A time stamp in UTC format specifying when this list was created.
**tp_LastDeleted**: A time stamp in UTC format specifying when an item was last deleted from this list. This value MUST default to NULL if no item has been deleted.

**tp_Version**: A counter incremented any time a change is made to the schema or other properties of this list and used for internal conflict detection.

**tp_BaseType**: The **List Base type** of this list. The following values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&quot;Generic&quot; list</td>
</tr>
<tr>
<td>1</td>
<td>Document library</td>
</tr>
<tr>
<td>3</td>
<td>&quot;Discussion board&quot; list</td>
</tr>
<tr>
<td>4</td>
<td>&quot;Survey&quot; list</td>
</tr>
<tr>
<td>5</td>
<td>&quot;Issues&quot; list</td>
</tr>
</tbody>
</table>

**tp_FeatureId**: The Feature Identifier for the feature that defines the base schema of this list.

**tp_ServerTemplate**: The **list template identifier** that defines the base structure of this list.

**DirName**: The directory name of the location that contains this list.

**LeafName**: The leaf name of the location that contains this list.

**DirName**: The directory name of the default **document template** in the list. This value can be NULL if a document template is not defined for this list.

**LeafName**: The leaf name of the default document template in the list. This value can be NULL if a document template is not defined for this list.

**tp_ReadSecurity**: This signifies special restrictions that can be placed on list item access. The following values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No special restrictions.</td>
</tr>
<tr>
<td>2</td>
<td>Users can see only their own list items. The front-end Web server MUST NOT display list items to users without the <code>ManageLists</code> permissions unless the list item was created by that user (for example, <code>tp_Author</code> = <code>@UserId</code>).</td>
</tr>
</tbody>
</table>

**tp_WriteSecurity**: This signifies special restrictions that can be placed on list item update. The following values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No special restrictions.</td>
</tr>
<tr>
<td>2</td>
<td>Users will see only their own list items. The front-end Web server MUST NOT allow users without the <code>ManageLists</code> permission to update a list item unless the list item was created by that user (for example, <code>tp_Author</code> = <code>@UserId</code>).</td>
</tr>
<tr>
<td>4</td>
<td>Users will not update any list items in this list. The front-end Web server MUST NOT allow users without the <code>ManageLists</code> permission to add or update list items in this list.</td>
</tr>
</tbody>
</table>

**tp_Description**: The description of this list for display in the front-end Web server.
\{tp\_Fields\}: This field MUST be NULL if the site (2) or list has been flagged to cache all schema data; otherwise, it contains a Compressed structure. Uncompressed it contains an implementation-specific version string followed by an XML representation of the field definitions. The field definitions include display and interaction options. The XML MUST conform to the FieldDefinitionDatabaseWithVersion complex type, as specified in section 2.2.7.3.5.

**tp\_Direction**: An enumerated value specifying the direction of text flow for front-end Web server elements presented by this list. The following values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No explicit direction is specified.</td>
</tr>
<tr>
<td>1</td>
<td>Text flow is left to right.</td>
</tr>
<tr>
<td>2</td>
<td>Text flow is right to left.</td>
</tr>
</tbody>
</table>

**AnonymousPermMask**: A WSS Rights Mask that indicates the permissions granted to a user that is anonymous, or has no specific permissions, on this list.

**tp\_Flags**: A List Flags value describing this list.

**tp\_ThumbnailSize**: The width, in pixels, specified for use when creating thumbnail images of list items within this list. This value MUST be NULL for lists that do not have a List Server Template value of 109(Image Library Template). Thumbnail images are generated by the front-end Web server for documents and have implementation-specific capabilities.

**tp\_WebImageWidth**: The width, in pixels, specified for use when creating Web images of list items within this list. This value MUST be NULL for lists that do not have a List Server Template value of 109(Image Library Template). Web images are generated by the front-end Web server for documents and have implementation-specific capabilities.

**tp\_WebImageHeight**: The height, in pixels, specified for use when creating Web images of list items within this list. This value MUST be NULL for lists that do not have a List Server Template value of 109(Image Library Template).

**tp\_ImageUrl**: The URL of the image used to represent this list.

**tp\_ItemCount**: The number of list items that are stored within this list.

**tp\_Author**: The identifier of the user who is listed as creating this list.

**tp\_HasInternalFGP**: This flag is set to 1 if there have ever been list items for this list that have had a unique access control list (ACL) applied. Otherwise, this flag is set to 0.

**tp\_ScopeId**: Unique identifier of the security scope for this list. This indicates the specific WSS ACL Format access control list (ACL) to use for calculating the permission settings on this list.

**Acl**: The binary serialization of the WSS ACL Format access control list (ACL) for this list. This MUST be used for the permissions for this list if the tp\_ScopeId value of this list is set to unique security scope.

**tp\_EventSinkAssembly**: The name of the .NET assembly that implements the event sink associated with this list. This value MUST default to NULL if no event sink has been associated with the list.

**tp\_EventSinkClass**: The name of the .Net assembly class that implements the event sink associated with this list. This value MUST default to NULL if no event sink has been associated with the list.
tp_EventSinkData: Additional data persisted on behalf of the event sink implementation, to be passed to the event sink associated with this list. This value MUST default to NULL if no event sink has been associated with the list.

tp_EmailAlias: The e-mail address of the list. This e-mail address is used to allow files to be sent directly to this list through an implementation-specific e-mail handling feature. This value MUST default to NULL if no e-mail inserts server has been associated with the list.

tp_WebFullUrl: The complete store-relative form URL to the site (2) that contains this list.

tp_WebId: The Site Identifier of the site (2) that contains the list.

tp_WebTitle: The title, for display in the front-end Web server, of the site (2) that contains this list.

tp_Web: The identifier of the site template for the site (2) that contains this list.

tp_WebLanguage: The language code identifier (LCID) of the display language of the site (2) that contains this list.

tp_WebCollation: The collation order for information in the site (2) that contains this list.

tp_SendToLocation: Contains an implementation-specific string of the URL used to copy list items to alternative locations. This value MUST be NULL if no "Send To Location" has been associated with the list.

{tp_MaxMajorVersionCount}: If the list has versioning enabled, this field contains the number of major versions that will be retained for this document. All versions more than tp_MaxMajorVersionCount removed from the current version of the document are automatically removed at version creation time. A value of 0 specifies that versions aren't automatically removed for this list.

{tp_MaxMajorwithMinorVersionCount}: If the list has versioning enabled, this field contains the number of major versions that will have their associated minor versions retained for this document. All versions more than tp_MaxMajorVersionCount removed from the current version of the document are automatically removed at version creation time. A value of 0 specifies that versions aren't automatically removed for this list.

tp_MaxRowOrdinal: This specifies the maximum row ordinal used to store list items for this list. This value indicates an implementation-specific calculation for storage of list items within lists.

tp_ListDataDirty: This is set to 1 if the list items in this list require dependency update processing before their next access (for example, updating document link information by parsing each document).

tp_DefaultWorkflowId: The Workflow Identifier corresponding to the workflow (2) invoked if the document is in a moderated list and the document is submitted for approval as part of a check in. If the document does not exist, or is not contained in a list with a configured approval workflow (2), this value MUST be NULL.

{HasBackLookupRels}: This bit is set to 1 if this list has any relationship lookup fields with relationship delete behavior that cascades or relationship delete behavior that is restricted. Otherwise, it MUST be 0.

tp_ContentTypes: Compressed structure. Uncompressed it will get XML data specifying the content types registered for this list.

tp_Subscribed: Set to 1 if an alert for changes to this list has been created in the past, signifying that additional processing needs to be performed.
{ChildCount}: If the document is contained within a list, this MUST be the sum total of list items and folders within the list. Otherwise, this MUST be the sum total of documents and folders that are contained within directory specified by the DirName column.

tp_NoThrottleListOperations: If this list is exempt from resource throttling operations, then this value MUST be 1. Otherwise, it MUST be 0.

tp_DescriptionResource: The resource token or resource identifier of the description for the site (2).

tp_ValidationFormula: A string representing the data validation criteria used to perform custom validation rules prior to the list being updated.

Followable: If this list is followable, then this value MUST be 1. Otherwise, it MUST be 0.

tp_ValidationMessage: A string containing text to display in the user interface when the list fails validation based on the data validation criteria represented by tp_ValidationFormula.

2.2.4.15 List Web Parts Result Set

The List Web Parts Result Set contains information about the Web Parts related to the lists associated with a specified document.

The List Web Parts Result Set is defined using T-SQL syntax, as follows.

tp_ListID uniqueidentifier,
tp_Type tinyint,
tp_Id uniqueidentifier,
tp_Flags int,
tp_DisplayName nvarchar(255),
tp_PageUrl nvarchar(255),
tp_BaseViewId tinyint,
tp_View varbinary(max),
tp_Level tinyint,
tp_ContentTypeId varbinary(512),
tp_PageUrlID uniqueidentifier,
tp_AllUsersProperties varbinary(max),
tp_PerUserProperties varbinary(max),
tp_WebPartIdProperty nvarchar(255),
tp_Cache varbinary(max)

tp_ListID: The list identifier of the list containing the Web Part.

tp_Type: The Page Type (section 2.2.1.2.14) of the Web Part.

tp_Id: The Web Part Identifier (section 2.2.1.1.15) of the Web Part.

tp_Flags: A View Flags (section 2.2.2.13) value specifying view-related settings for this Web Part.

tp_DisplayName: The display name specified for the Web Part, if any. This value can be an empty string.

tp_PageUrl: The store-relative form URL to the site (2) that contains this Web Part.

tp_BaseViewId: The View Identifier (section 2.2.1.1.14) for the view where this Web Part is used.

tp_View: Contains implementation-specific XML used when processing this Web Part. This data is stored as a Compressed Structure (section 2.2.3.8), and is compressed by the algorithm specified in [RFC1950].
**tp_Level**: A Publishing Level Type (section 2.2.6) value specifying the publishing status of the Web Part. This value MUST be NULL if the Web Part does not exist.

**tp_ContentTypeId**: A binary content type identifier value specifying the valid content types that this Web Part can be used to view. If this Web Part is not restricted to a particular content type, this value MUST be zero.

**tp_PageUrlID**: The document identifier with which the Web Part is associated.

**tp_AllUsersProperties**: A list of the XML properties which are common for all users of the Web Part. This value can be NULL.

**tp_PerUserProperties**: A list of the XML properties which are specific to a particular user of the Web Part. This value can be NULL.

**tp_WebPartIdProperty**: A string which identifies the Web Part within the page.

**tp_Cache**: The private data that is cached for the Web Part. This value can be NULL.

### 2.2.4.16 NULL Individual URL Security Result Set

The NULL Individual URL Security Result Set indicates that a specified document is not found.

The NULL Individual URL Security Result Set MUST return a single row and is defined using T-SQL syntax, as follows.

```sql
{ListId}                   uniqueidentifier = NULL,
{Acl}                      varbinary(max) = NULL,
{AnonymousPermMask}        bigint = NULL,
{IsAttachment}             bit = 0,
{NeedManageListRight}      bit = 0,
{BaseType}                 int = NULL,
{ExcludedType}             int = NULL,
{ListFlags}                bigint = NULL,
{Level}                    tinyint = NULL,
{DraftOwnerId}             int = NULL,
{DocType}                  tinyint = NULL,
{bNeedNoThrottle}          bit = NULL,
{ItemCount}                int = NULL,
{DoclibRowId}              int = 0,
{DocFlags}                 int = 0;
```

**{ListID}**: MUST be NULL.

**{Acl}**: MUST be NULL.

**{AnonymousPermMask}**: MUST be NULL.

**{IsAttachment}**: MUST be 0.

**{NeedManageListRight}**: MUST be 0.

**{BaseType}**: MUST be NULL.

**{ExcludedType}**: MUST be NULL.

**{ListFlags}**: MUST be NULL.

**{Level}**: MUST be NULL.

**{DraftOwnerId}**: MUST be NULL.
{DocType}: MUST be NULL.

{bNeedNoThrottle}: MUST be NULL.

{ItemCount}: MUST be NULL.

{DoclibRowId}: MUST be 0.

{DocFlags}: MUST be 0.

2.2.4.17 NULL Unique Permissions Result Set

The NULL Unique Permissions Result Set indicates that unique security scopes are not found in the specified location. When returned, the result set MUST contain a single row of data. The NULL Unique Permissions Result Set is defined using T-SQL syntax, as follows.

```sql
{ScopeId} binary(1),
{Acl} varbinary(max),
{AnonymousPermMask} binary(1);
```

{ScopeId}: This MUST be 0x00. This field can be cast as a uniqueidentifier with a value of 00000000-0000-0000-0000-000000000000.

{Acl}: This MUST be NULL.

{AnonymousPermMask}: This MUST be 0x00. This field can be cast as a bigint.

2.2.4.18 Object ID Result Set

The Object ID Result Set contains the GUID of the matching object, if any. The Object ID Result Set is defined using T-SQL syntax, as follows.

```sql
Id uniqueidentifier
```

Id: Contains the GUID of the matching object.

2.2.4.19 Principal User Information Result Set

The Principal User Information Result Set returns information about a principal (1). The Principal User Information Result Set is defined using T-SQL syntax, as follows.

```sql
tp_Id int,
.tp_SystemID varbinary(512),
.tp_Title nvarchar(255),
.tp_Login nvarchar(255),
.tp_Email nvarchar(255),
.tp_Notes nvarchar(1023),
.tp_SiteAdmin bit,
.tp_DomainGroup bit,
.tp_Flags int;
```

tp_Id: The user identifier of the principal. MUST NOT be NULL.

tp_SystemID: The SystemId of the principal. This parameter MUST NOT be NULL.

tp_Title: Contains the display name of the principal. This parameter MUST NOT be NULL.
tp_Login: Contains the login name of the principal. This parameter MUST NOT be NULL.

tp_Email: Contains the e-mail address of the principal. This parameter MUST NOT be NULL.

tp_Notes: Contains notes associated with the principal. This parameter MUST NOT be NULL.

tp_SiteAdmin: Set to 1 if the principal is a site collection administrator; otherwise 0. This parameter MUST NOT be NULL.

tp_DomainGroup: Set to 1 if the principal is a domain group; otherwise 0. This parameter MUST NOT be NULL.

tp_Flags: The User Information flags value of the principal. This parameter MUST NOT be NULL.

2.2.4.20 Server Time Result Set

The Server Time Result Set returns the current time from the back-end database server in UTC. The Server Time Result Set is defined using T-SQL syntax, as follows.

```sql
{CurrentTime}       datetime;
```

CurrentTime: The current time from the back-end database server, in UTC format.

2.2.4.21 Single Doc Link Information Result Set

The Link Info Result Set returns information about each forward link from, and each backward link to, a specified document or document version.

The Link Info Result Set MUST be ordered by the DocId column and is defined using T-SQL syntax, as follows.

```sql
DocId          uniqueidentifier,
LinkDirName   nvarchar(256),
LinkLeafName  nvarchar(128),
LinkType      tinyint,
LinkSecurity  tinyint,
LinkDynamic   tinyint,
LinkServerRel bit,
LinkStatus    tinyint,
PointsToDir   bit,
WebPartId     uniqueidentifier,
LinkNumber    int,
WebId         uniqueidentifier,
Search        nvarchar(max),
FieldId       uniqueidentifier;
```

DocId: The document identifier of the specified document.

LinkDirName: Contains the directory name of the link target.

LinkLeafName: Contains the leaf name of the link target.

LinkType: The LinkType value of the link. See LinkInfo Types (section 2.2.1.2.10) for a list of valid LinkType values. This value MUST be NULL for a backward link.

LinkSecurity: The LinkSecurity value of the forward link, which specifies whether the scheme of the link is HTTP or HTTPS. This value MUST be NULL for a backward link.
LinkDynamic: The **LinkDynamic** value of the forward link, which specifies whether the link is to a special type of target. This value MUST be NULL for a backward link.

LinkServerRel: A bit flag which specifies whether the link URL is server-relative or not. A value of one specifies a server-relative URL. This value MUST be NULL for a backward link.

LinkStatus: The **Document Store Type** value (section 2.2.4) of the document targeted by a forward link. This value MUST be 128 for a backward link. If the forward link target is a document that does not exist, or if this link refers to a target that exists outside the specified site collection, or if this link refers to a location that could not be verified, this value MUST be NULL.

PointsToDir: A bit flag specifying whether the target of the forward link was a directory and has been modified to target a welcome page. This value MUST be NULL for a backward link. For a forward link, if the link target is a directory where a welcome page is specified, the link MUST be changed to the URL of the welcome page and **PointsToDir** MUST be set to one so that the link can be distinguished from an explicit link to the welcome page; otherwise, this value MUST be zero.

WebPartId: This value MUST be NULL.

LinkNumber: This value MUST be NULL.

WebId: This value MUST be NULL.

Search: This value MUST be NULL.

FieldId: This value MUST be NULL.

### 2.2.4.22 Site Audit Mask Result Set

The **Site Audit Mask Result Set** contains the context-sensitive identifier for a specified object, and the **Audit Flags** set and inherited on that object. The **Site Audit Mask Result Set** is defined using T-SQL syntax, as follows.

```sql
{Id} uniqueidentifier,
{AuditFlags} int,
{InheritAuditFlags} int,
{SiteGlobalAuditMask} int;
```

- **{Id}**: The identifier for the object. The **Audit Item Type** of the object determines the type of identifier used for this value (for example, a Site Identifier or a List Identifier). If an object of the specified **Audit Item Type** at the specified location is found, the value MUST NOT be NULL; otherwise, it MUST be NULL.

- **{AuditFlags}**: An **Audit Flags** value specifying the operations to be audited that are set directly on the specified object. If an object of the specified **Audit Item Type** at the specified location is not found, the value MUST be NULL.

- **{InheritAuditFlags}**: An **Audit Flags** value specifying the operations to be audited on the specified object that are inherited from a contained document. If an object of the specified **Audit Item Type** at the specified location is not found, the value MUST be NULL.

- **{SiteGlobalAuditMask}**: An **Audit Flags** value specifying the operations to be audited on the specified object that are applied globally within the specified site collection.

### 2.2.4.23 Site Feature List Result Set

The **Site Feature List Result Set** contains the list of Feature Identifiers of a **site** (2).
The **Site Feature List Result Set** is defined using T-SQL syntax, as follows.

**FeatureId** uniqueidentifier;

**FeatureId:** A Feature Identifier for a feature of a site (2).

### 2.2.4.24 Site Metadata Result Set

The **Site Metadata Result Set** contains metadata for a site (2) or sites.

The **Site Metadata Result Set** is defined using T-SQL syntax, as follows.

```
Id                             uniqueidentifier,
Title                          nvarchar(255),
Description                    nvarchar(max),
MetaInfoVersion                int,
WebTemplate                    int,
Language                       int,
Locale                         int,
Collation                      smallint,
TimeZone                       smallint,
Time24                         bit,
CalendarType                   smallint,
AdjustHijriDays                smallint,
AltCalendarType                tinyint,
CalendarViewOptions            tinyint,
WorkDays                       smallint,
WorkDayStartHour               smallint,
WorkDayEndHour                 smallint,
ProvisionConfig                smallint,
Flags                          int,
Author                         int,
AlternateCSSUrl                nvarchar(260),
CustomizedCss                  nvarchar(260),
CustomJSUrl                    nvarchar(260),
AlternateHeaderUrl             nvarchar(260),
SecurityProvider               uniqueidentifier,
MasterUrl                      nvarchar(260),
CustomMasterUrl                nvarchar(260),
{SiteLogoUrl}                  nvarchar(260),
{SiteLogoDescription}          nvarchar(255),
AllowMUI                       bit,
TitleResources                 nvarchar(256),
AlternateMUICultures           nvarchar(max),
OverwriteMUICultures           bit,
UIVersion                      tinyint,
ClientTag                      smallint,
AnonymousPermMask              bigint,
{SiteFlags}                    int,
{SitePortalURL}                nvarchar(260),
{SitePortalName}               nvarchar(255),
MeetingCount                   smallint,
DefTheme                       nvarchar(64),
CachedNav                      varbinary(max),
CachedInheritedNav             varbinary(max),
CachedNavDirty                  int,
CachedDataVersion              int,
NavParentWebId                  uniqueidentifier,
FirstUniqueAncestorWebId      uniqueidentifier,
ScopeId                        uniqueidentifier,
DbNow                          datetime,
Acl                            varbinary(max),
{RequestAccessEmail}           nvarchar(255),
Ancestry                       varbinary(max),
```
ProductVersion        smallint,
tp_Id                  int,
tp_SiteAdmin           bit,
tp_IsActive            bit,
tp_Login               nvarchar(255),
tp_Email               nvarchar(255),
tp_Title               nvarchar(255),
tp_Notes               nvarchar(1023),
tp_ExternalTokenLastUpdated    datetime,
tp_Token               varbinary(max),
tp_Flags               int,
UserId                 int,
{SiteSecurityVersion}  bigint,
tpLocale               int,
tp_TimeZone            smallint,
tp_Time24              bit,
tp_CalendarType        smallint,
tp_Places              bigint,
tp_CalendarViewOptions tinyint,
tp_CalendarViewOptions tinyint,
tp_WorkDays            smallint,
tp_WorkDayStartHour    smallint,
tp_WorkDayEndHour      smallint,
tp_MUILanguages        varchar(64),
tp_ContentLanguages    varchar(64),
{SiteHashKey}          binary(16),
{UserInfoListId}       uniqueidentifier,
{RootWebId}            uniqueidentifier,
{SiteLastContentChange} datetime,
{SiteLastSecurityChange} datetime,
{RbsCollectionId}      int,
{DenyPermMask}         tPermMask;

**Id:** The Site Identifier for the site (2).

**Title:** The title of the site (2) for display in the front-end Web server.

**Description:** The description of the site (2) for display in the front-end Web server.

**MetaInfoVersion:** A counter incremented any time a change is made to the Metainfo for this site (2) and used for internal conflict detection.

**WebTemplate:** The identifier for the site template used in the site definition to define the base structure of this site (2).

**Language:** The LCID associated with the site (2) which is used to determine current UI culture, which determines which language resources to use to display messages on the front-end Web server.

**Locale:** The LCID associated with the site (2) which is used to determine the current culture for regional language specific data formatting such as currency or date and time settings.

**Collation:** The Collation Order of the site (2) which indicates an additional sorting order that is processed by the back-end database server. The collation method is an implementation-specific capability of the front-end Web server and back-end database server.

**TimeZone:** The Time Zone Identifier (section 2.2.1.3) for the time zone to be used when displaying time values for this site (2).

**Time24:** If set to "1", use a 24-hour time format when displaying time values for this site; otherwise, a 12-hour time format can be used.

**CalendarType:** The Calendar Type that is to be used when processing date values for this site (2).
AdjustHijriDays: If the Calendar Type value is "6", this specifies the number of days to extend or reduce the current month in Hijri calendars for this site (2).

AltCalendarType: The Calendar Type of an alternate calendar for processing date values for this site (2). If NULL, only the CalendarType value is used for this site (2).

CalendarViewOptions: A Calendar View Options Type that specifies the calendar display options setting for this site (2).

WorkDays: A set of Workdays Flags that specify the weekdays defined as the workweek for this site (2).

WorkDayStartHour: The start time of the workday in minutes after midnight for this site (2).

WorkDayEndHour: The end time of the workday in minutes after midnight for this site (2).

ProvisionConfig: An identifier of the site template used to provision this site (2). The following reserved site template identifiers are defined.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>This site (2) has not had any site template provisioned.</td>
</tr>
<tr>
<td>0</td>
<td>This site (2) has the implementation-specific default site template applied.</td>
</tr>
<tr>
<td>1</td>
<td>This site (2) has the Team Collaboration site template applied.</td>
</tr>
<tr>
<td>2</td>
<td>This site (2) has the Meeting Workspace site template applied.</td>
</tr>
<tr>
<td>3</td>
<td>This site (2) has the Central Administration site template applied.</td>
</tr>
<tr>
<td>4</td>
<td>This site (2) has the Wiki site template applied.</td>
</tr>
<tr>
<td>9</td>
<td>This site (2) has the Blog site template applied.</td>
</tr>
</tbody>
</table>

Flags: A Site Property Flags value describing the configuration of this site (2).

Author: The User Identifier of the user who is listed as creating this site (2).

AlternateCSSUrl: The URL for a custom CSS sheet file registered on the site (2) for use in pages of the site.

CustomizedCss: This contains an implementation-specific list of custom CSS files associated with this site (2).

CustomJSUrl: The URL for a custom JavaScript file registered on the site (2) for use in pages of the site.

AlternateHeaderUrl: The URL for a custom header HTML page registered on the site (2) for use in pages of the site when rendered on the front-end Web server.

SecurityProvider: COM class identifier (CLSID) of the external security provider for this site (2). This is NULL for sites using the native security implementation.

MasterUrl: The URL for the master page registered on the site (2) for use in pages of the site when rendered on the front-end Web server.

CustomMasterUrl: The URL for an alternate master page registered on the site (2) for use in pages of the site rendered on the front-end Web server.

{SiteLogoUrl}: The URL of an image that represents the site (2) for display in the front-end Web server.
{SiteLogoDescription}: The description of the image that represents the site (2) for display in the front-end Web server as an ALT tag on the image.

@AllowMUI: A bit which indicates whether the Multilingual User Interface feature is enabled.

@TitleResource: The resource token or resource identifier of the title for the site (2) whose metadata is to be updated.

@DescriptionResource: The resource token or resource identifier of the description for the site (2) whose metadata is to be updated.

@AlternateMUICultures: The string that contains the distinct language identifier(s) for all the alternate language(s) of the site (2). For example, the language identifier for English is 1033. Every element is separated by semicolon from the next.

@OverwriteMUICultures: A bit which specifies whether the changes made to user-specified text in the default language automatically overwrite the existing translations made in all alternate languages.

@UIVersion: A number that represents the visual state of the site (2).

@ClientTag: A number that represents the state of the application files of the site (2).

AnonymousPermMask: A WSS Rights Mask that indicates the permissions granted to a user that is anonymous, or has no specific permissions, to this site (2). The value MUST be NULL for sites that do not exist.

{SiteFlags}: A Site Collection Flags value that indicates the settings for the site collection that contains this site (2).

{SitePortalURL}: The URL for a different site (2) registered on the site for use in navigation structures as a parent location.

{SitePortalName}: The display name of a different site (2) registered on the site for use in navigation structures as a parent location.

MeetingCount: If this site (2) is a meeting workspace (that is, its site template identifier in ProvisionConfig is set to 2), this value indicates the number of meetings that are configured.

DefTheme: The name of a theme that is used as part of the display of the site (2).

CachedNav: An implementation-specific serialization of navigation structure information to display in front-end Web server elements associated with this site (2).

CachedInheritedNav: This contains an implementation-specific serialization of navigation structure information to display in front-end Web server elements associated with the root navigation elements for this site (2).

CachedNavDirty: If this value is not set to "0", the cached navigation information for this site MUST be regenerated.

CachedDataVersion: A counter incremented on every change to the cached navigation information, used for internal conflict detection.

NavParentWebId: The Site Identifier of the site (2) to be treated as the parent of this site when displaying navigation elements in the front-end Web server.

FirstUniqueAncestorWebId: The Site Identifier of the closest site (2) in this site's ancestor chain that does not inherit security settings from its parent site.

ScopeId: The Scope Identifier of the security scope containing the site (2).
DbNow: The current time in UTC format on the back-end database server.

Acl: The binary serialization of the WSS ACL Format ACL for this site (2). This is either explicitly defined or inherited from the parent object of the site (2).

{RequestAccessEmail}: The e-mail address for a user who has the authority to grant access to the site (2), if any. This can be used by the front-end Web server to generate an e-mail to request access for a user who does not have access to the site (2).

Ancestry: An implementation-specific serialization of the navigation structure information for the site (2).

ProductVersion: The implementation-specific version identifier used to create the site (2).

tp_Id: The User Identifier of the current user.

tp_SiteAdmin: Set to "1" if the current user is an administrator of the site (2).

tp_IsActive: Set to "1" if the current user is an active user in the site collection containing this site (2).

tp_Login: The login information of the current user.

tp_Email: The e-mail address of the current user.

tp_Title: The display name of the current user.

tp_Notes: Notes about the current user.

tp_ExternalTokenLastUpdated: A time stamp in UTC format specifying the time when the External Group Token for the current user was last updated.

tp_Flags: A WSS User Flags value of the specified user

tp_Token: A WSS User Token value specifying the site group membership of the current user.

UserId: The site membership identifier of the current user. This is NULL if the current user has not been added to the set of members of this site (2).

{SiteSecurityVersion}: The current security information version of the site collection.

tp_Locale: An LCID referring to the locale value to be used when displaying messages in the front-end Web server for the current user. If this value is NULL, the site setting for Locale is used instead.

tp_TimeZone: The Time Zone Identifier for the time zone to be used when displaying time values for the current user. If this value is NULL, the site setting for TimeZone is used instead.

tp_Time24: A bit flag which specifies whether to use a 24-hour time format when displaying time values to the current user. If this parameter is set to "1", the 24-hour time format is be used; otherwise, the 12-hour time format is be used. If this value is NULL, the site setting for Time24 MUST be used instead.

tp_CalendarType: The Calendar Type to be used when processing date values for the current user. If this value is NULL, the site setting for CalendarType MUST be used instead.

tp_AdjustHijriDays: If the tp_CalendarType value is "6", this specifies the number of days to extend or reduce the current month in Hijri calendars for the current user. If this value is NULL, the site setting for AdjustHijriDays MUST be used instead.

tp_AltCalendarType: The Calendar Type of an alternate calendar to be used when displaying calendars in views for the current user. If this value is NULL, the site setting for AltCalendarType MUST be used instead.
tp_CalendarViewOptions: A Calendar View Options Type specifying the calendar options for the current user. If this value is NULL, the site setting for CalendarViewOptions MUST be used instead.

tp_WorkDays: A set of Workdays Flags specifying the weekdays defined as the workweek for the current user. If this value is NULL, the site setting for WorkDays MUST be used instead.

tp_WorkDayStartHour: The start time of the work day in minutes after midnight for the current user. If this value is NULL, the site setting for WorkDayStartHour MUST be used instead.

tp_WorkDayEndHour: The end time of the workday in minutes after midnight for the current user. If this value is NULL, the site setting for WorkDayEndHour MUST be used instead.

tp_MUILanguages: A string that contains the distinct culture(s) for the user's preferred display language(s), separated by semicolon. This string can be NULL.

tp_ContentLanguages: A string that contains the distinct culture(s) for the user's preferred content language(s), separated by semicolon. This string can be NULL.

{SiteHashKey}: Contains binary information used when generating digital signatures of information associated with this site collection.

{UserInfoListId}: The list identifier of the user information list for the site collection containing this site (2).

{RootWebId}: The site identifier for the root site of the site collection containing this site (2).

{SiteLastContentChange}: A time stamp in UTC format describing the time content was last changed within this site collection.

{SiteLastSecurityChange}: A time stamp in UTC format describing the time security settings were last changed within this site collection.

{RbsCollectionId}: The identifier for the remote BLOB storage collection for the site collection, or zero if remote BLOB storage is not configured for this database.

{DenyPermMask}: A WSS Rights Mask (section 2.2.15) specifying the rights to deny to the current user.

2.2.4.25 Site MetaInfo Result Set

The Site MetaInfo Result Set returns metainfo about a site (2).

The Site MetaInfo Result Set is defined using T-SQL syntax, as follows.

    MetaInfo          varbinary(max);

MetaInfo: Site MetaInfo in the form of a metadict. This value can be NULL. The metadict format is specified in the [MS-FPSE] metadict section.

2.2.4.26 Unique Permissions Result Set

The Unique Permissions Result Set returns the ACL information for all the unique security scopes of folders and list items contained in a specified location.

The Unique Permissions Result Set is defined using T-SQL syntax, as follows.

    ScopeId                  uniqueidentifier,
    Acl                      varbinary(max),
AnonymousPermMask: A WSS Rights Mask that indicates the permissions granted to an anonymous user, or a user who has no specific permissions on this security scope.

2.2.4.27 URL Result Set

The URL Result Set returns a list of URLs, one for the root folder of each site (2). The URL Result Set is defined using T-SQL syntax as follows.

```
FullUrl                        nvarchar (256)
```

FullUrl: The store-relative form URL of the site (2).

2.2.4.28 List Related Fields Result Set

The List Related fields Result Set returns information about all the relationship lookup fields whose target list is the specified list. The T-SQL syntax for this result set is as follows:

```
 tp_WebId                  uniqueidentifier NOT NULL,
 tp_Id                     uniqueidentifier NOT NULL,
 FieldId                   uniqueidentifier NOT NULL,
 DeleteBehavior            tinyint NOT NULL
```

tp_WebId: The site identifier of the site (2) which contains the relationship lookup field.

tp_Id: The list identifier of the list which contains the relationship lookup field.

FieldId: The field identifier of the relationship lookup field.

DeleteBehavior: The relationship delete behavior on the relationship lookup field.

2.2.4.29 App Principal Information Result Set

The App Principal Information Result Set returns information about an app principal.

If the App Principal Info Result Set is returned, it MUST return one row, as defined using the T-SQL syntax below:

```
{AppPrincipalName}             nvarchar(256),
{ReturnAppPrincipalInfo}       bit,
{WebTemplateId}                int,
{WebTemplateConfigurationId}   int
```

{AppPrincipalName}: The app principal identifier.

{ReturnAppPrincipalInfo}: This value MUST be 1.

{WebTemplateId}: The identifier for the site template used in the site definition associated with the stored procedure.
2.2.4.30 App Principal Permissions Result Set

The **App Principal Permissions Result Set** is returned to indicate what permissions the specified **app principal** has.

If the **App Principal Permissions Result Set** is returned, it MUST return at least one row, as defined using the T-SQL syntax below:

```sql
{Id}           int,
{Name}         nvarchar(256),
{Title}        nvarchar(256),
{Flag}         int,
{WebId}        uniqueidentifier,
{ListId}       uniqueidentifier,
{Perm}         int
```

*{Id}*: The **app principal** integer identifier in the site collection.

*{Name}*: The **app principal** identifier.

*{Title}*: The title of the **app principal**.

*{Flag}*: The **app principal** flag that specifies the state of the app principal. All valid values for this flag are specified in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No flags.</td>
</tr>
<tr>
<td>1</td>
<td>The app principal has been disabled.</td>
</tr>
<tr>
<td>2</td>
<td>The app principal has no tenant scoped permissions.</td>
</tr>
<tr>
<td>4</td>
<td>Allow App Only Policy.</td>
</tr>
</tbody>
</table>

*{WebId}*: The site identifier of the site (2) containing the document.

*{ListId}*: The list identifier of the list containing the document.

*{Perm}*: The permission value as defined in section 2.2.1.2.18.

2.2.5 SQL Structures

2.2.5.1 Configuration Object

A Configuration Object encapsulates as a group of application settings a set of metadata used to identify and manipulate those settings. Configuration Objects have the following fields.

2.2.5.1.1 Class

Applications use a Configuration Object **Class** to help distinguish the schemas of the data stored in different objects. **Class** is a complex data type with the following fields, defined in T-SQL syntax.

```sql
ClassId                        uniqueidentifier NOT NULL,
BaseClassId                    uniqueidentifier NOT NULL,
```
**ClassId**: This MUST be a GUID that is different from all other ClassIds registered in the Configuration Database.

**BaseClassId**: A class is said to derive from the class referenced by BaseClassId. This allows a class hierarchy to be built. Many ClassIds can share a single BaseClassId. The BaseClassId of a Configuration Object Class MUST have the same value as a ClassId in the Configuration Database. If a Configuration Object Class has no base class, its BaseClassId MUST be identical to its ClassId.

**FullName**: An identifier that can be used by an application to associate a ClassId with a human-readable or machine-readable string.

The following classes are used during the execution of this protocol.

<table>
<thead>
<tr>
<th>Class</th>
<th>ClassId</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate URL Collection</td>
<td>9920F4B6-2FF4-4d10-9532-E01979826585</td>
</tr>
<tr>
<td>Content Database</td>
<td>3D4F5451-1735-48bb-B920-76C1EC240B1D</td>
</tr>
<tr>
<td>Database Service Instance</td>
<td>3112E92F-B97D-4B1e-8CEB-03FDE1ED1A7</td>
</tr>
<tr>
<td>Farm</td>
<td>674D5AA5-EA77-4A3-B9F8-3F70D786DE6A</td>
</tr>
<tr>
<td>Server</td>
<td>E77AAF47-3CAC-4001-BC6B-5BCCB6486318</td>
</tr>
<tr>
<td>Web Application</td>
<td>113F5B89-7520-4651-8FC4-E9F4F887618</td>
</tr>
<tr>
<td>Web Service</td>
<td>45AD2BF2-4E3E-46A1-B477-126944C0ACEF</td>
</tr>
</tbody>
</table>

### 2.2.5.1.2 Id

A Configuration Object's Id is used to identify a single instance of a Configuration Object. As such, it MUST be different from the IDs of all other Configuration Objects in the Configuration Database. Id is defined in T-SQL format as follows.

```
Id                             uniqueidentifier NOT NULL
```

### 2.2.5.1.3 Name

Configuration Objects can also be identified by name. Unlike Id, a Configuration Object's name is only used to distinguish instances of Configuration Objects with the same parent and class. The combination of name, parent, and class MUST be unique among the set of Configuration Objects in a single Configuration Database. Name is defined in T-SQL format as follows.

```
Name                      nvarchar(128) NOT NULL
```

### 2.2.5.1.4 Parent

Configuration Objects are organized in an ancestry hierarchy. Each Configuration Object MUST have one and only one parent, which MUST be a Configuration Object registered in the same Configuration Database. The set of Configuration Objects with a given parent are known as the parent's children. A
Configuration Object's descendants are the set of Configuration Objects including the parent's children, the children's children, and so on.

The Configuration Object at the root of an ancestry hierarchy MUST be its own parent.

The Configuration Database MUST delete a Configuration Object when its parent is deleted.

A Configuration Object defines its parent through its own ParentId property, which is the Id of its parent. ParentId is defined in T-SQL format as follows.

```
ParentId                       uniqueidentifier NOT NULL
```

### 2.2.5.1.5 Status

The Status property of a Configuration Object MUST conform to the data type defined in the Configuration Object Status section.

```
Status         int NOT NULL
```

### 2.2.5.1.6 Version

Version is used to identify when a Configuration Object has been created, modified, or deleted. Version is defined in T-SQL format as follows.

```
Version                        rowversion NOT NULL
```

### 2.2.5.1.7 Properties

Any configuration settings needed by an application that are not stored in one of the other Configuration Object field can be stored in the Properties field, which is defined using T-SQL format as follows.

```
Properties                     nvarchar(max)
```

Successful execution requires parsing the properties of the following classes. Each of these classes stores its properties in an XML format. Because only a portion of each class' Properties field is needed for successful execution, the values needed are identified using queries from the XML Path Language specified in [XPATH]. The schemata of the contents of other Configuration Objects' Property Bags will differ based on the needs of each application and are not specified in this document. Unless otherwise noted in the description column, each XPath query resolves to one and only one element value.

#### 2.2.5.1.7.1 Alternate URL Collection

An Alternate URL Collection Configuration Object stores a list of absolute URLs. The parent of an Alternate URL Collection MUST be a Farm Configuration Object.

<table>
<thead>
<tr>
<th>Property</th>
<th>XPath Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate URL</td>
<td>/object/fld[attribute::name='m_Urls']/fld/object/sFld[attribute::name='m_RequestUri']</td>
<td>When executed against the properties of an Alternate URL Collection</td>
</tr>
<tr>
<td>Property</td>
<td>XPath Query</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Username</td>
<td>/object/fld[attribute::name='m_Username']</td>
<td>If the value returned by this XPath query is not NULL or empty, the client uses SQL authentication to connect to the content database. The client MUST pass the value returned by the XPath query as the SQL authentication username.</td>
</tr>
<tr>
<td>Password</td>
<td>/object/fld[attribute::name='m_Password']</td>
<td>If the value returned by this XPath query is not NULL or empty, the client MUST include this value in the user name portion of the content database connection string.</td>
</tr>
</tbody>
</table>

### 2.2.5.1.7.2 Content Database

The **Content Database** Configuration Object stores information needed to establish a connection to a content database.

<table>
<thead>
<tr>
<th>Property</th>
<th>XPath Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>/object/fld[attribute::name='m_Username']</td>
<td>If the value returned by this XPath query is not NULL or empty, the client uses SQL authentication to connect to the content database. The client MUST pass the value returned by the XPath query as the SQL authentication username.</td>
</tr>
<tr>
<td>Password</td>
<td>/object/fld[attribute::name='m_Password']</td>
<td>If the value returned by this XPath query is not NULL or empty, the client MUST include this value in the user name portion of the content database connection string.</td>
</tr>
</tbody>
</table>

### 2.2.5.1.7.3 Web Application

The **Web Application** Configuration Object stores a list of content databases used by the **Web application** as well as information needed to parse URLs. The parent of a **Web Application** Configuration Object MUST be a **Web Service** Configuration Object.

<table>
<thead>
<tr>
<th>Property</th>
<th>XPath Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Application Alternate URL Collection</td>
<td>/object/fld[attribute::name='m_AlternateUrlCollection']</td>
<td>This value is a string representation of a Configuration Object ID. It is used to associate a Web application with an Alternate URL Collection.</td>
</tr>
</tbody>
</table>
### Property | XPath Query | Description
--- | --- | ---
Prefix Name | /object/fld[attribute::name='m_Prefixes']/object/fld/object/sFld[attribute::name='m_Name'] | Each Alternate URL Collection MUST be referenced by zero or one Web applications.
Prefix Type | /object/fld[attribute::name='m_Prefixes']/object/fld/object/sFld[attribute::name='m_Type'] | The value returned by this query MUST be a valid URL Path segment.

#### 2.2.5.2 Dependencies

In addition to the ancestry hierarchy defined in **Parent** (section 2.2.5.1.4), Configuration Objects can also define **Dependencies**. Dependencies can be used by an application to discover the list of Configuration Objects that depend on a given Configuration Object.

A **Dependency** is a complex type with the following fields, defined in T-SQL format as follows.

```sql
ObjectId uniqueidentifier NOT NULL
DependantId uniqueidentifier NOT NULL
```

**ObjectId**: The ID of the Configuration Object that depends on another.

**DependantId**: The ID of the Configuration Object upon which the Configuration Object identified by **ObjectId** depends.
2.2.6 Tables and Views

2.2.6.1 AllDocs Table

The AllDocs table stores data for all documents in the content database.

The AllDocs table is defined using T-SQL syntax as follows.

```sql
TABLE AllDocs(
    Id uniqueidentifier NOT NULL,
    SiteId uniqueidentifier NOT NULL,
    DirName nvarchar(256) NOT NULL,
    LeafName nvarchar(128) NOT NULL,
    WebId uniqueidentifier NOT NULL,
    ListId uniqueidentifier NULL,
    DoclibRowId int NULL,
    Type tinyint NOT NULL,
    SortBehavior tinyint NOT NULL DEFAULT 0,
    Size int NULL,
    ETagVersion AS
        WHEN InternalVersion IS NULL
            THEN NULL
        ELSE ((InternalVersion + (BumpVersion * 256)) / 256)
        END,
    EffectiveVersion AS
        WHEN InternalVersion IS NULL
            THEN NULL
        ELSE InternalVersion + (BumpVersion * 256)
        END,
    InternalVersion int NULL,
    BumpVersion tinyint NOT NULL DEFAULT 0,
    UIVersion int NOT NULL DEFAULT 512,
    Dirty AS CASE WHEN BumpVersion <> 0 THEN CAST(1 as bit) ELSE CAST(0 as bit) END,
    ListDataDirty bit NOT NULL DEFAULT 0,
    DocFlags int NULL,
    ThicketFlag bit NULL DEFAULT 0,
    CharSet int NULL,
    ProgId nvarchar(255) NULL,
    TimeCreated datetime NOT NULL,
    TimeLastModified datetime NOT NULL,
    NextToLastTimeModified datetime NULL,
    MetaInfoTimeLastModified datetime NULL,
    TimeLastWritten datetime NULL,
    DeleteTransactionId varbinary(16) NOT NULL DEFAULT 0x,
    SetupPathVersion tinyint NOT NULL DEFAULT 4,
    SetupPath nvarchar(255) NULL,
    SetupPathUser nvarchar(255) NULL,
    CheckoutUserId int NULL,
    CheckoutDate datetime NULL,
    CheckoutExpires datetime NULL,
    VersionCreatedSinceSTCheckout bit NOT NULL DEFAULT 0,
    LTCheckoutUserId AS
        CASE WHEN DocFlags & 32 = 32
            THEN CheckoutUserId
        ELSE NULL
        END,
    VirusVendorID int NULL,
    VirusStatus int NULL,
    VirusInfo nvarchar(255) NULL,
    VirusInfoEx varbinary(max) NULL,
    MetaInfo varbinary(max) NULL,
    MetaInfoSize int NULL,
    MetaInfoVersion int NOT NULL DEFAULT 1,
    UnVersionedMetaInfo varbinary(max) NULL,
    UnVersionedMetaInfoSize int NULL,
)
```
UnVersionedMetaInfoVersion int NULL,
WelcomePageUrl nvarchar(260) NULL,
WelcomePageParameters nvarchar(max) NULL,
IsCurrentVersion bit NOT NULL DEFAULT 1,
Level tinyint NOT NULL DEFAULT 1,
CheckinComment nvarchar(1023) NULL,
AuditFlags int NULL,
InheritAuditFlags int NULL,
DraftOwnerId int NULL,
UIVersionString AS CAST(UIVersion/512 AS nvarchar) + '.' + CAST(UIVersion%512 AS nvarchar),
ParentId uniqueidentifier NOT NULL,
HasStream AS
WHEN Type = 0 AND ((DocFlags & 256) = 256) AND SetupPath IS NULL OR (SetupPath IS NOT NULL AND ((DocFlags & 64) = 64))
THEN 1
ELSE 0
END,
ScopeId uniqueidentifier NOT NULL,
BuildDependencySet varbinary(max) NULL,
ParentVersion int NULL,
ParentVersionString AS CAST(ParentVersion/512 AS nvarchar) + '.' + CAST(ParentVersion%512 AS nvarchar),
TransformerId uniqueidentifier NULL,
ParentLeafName nvarchar(128) NULL,
IsCheckoutToLocal AS
CASE WHEN DocFlags & 512 = 512
THEN 1
ELSE 0
END,
CtoOffset smallint NULL,
Extension AS
CASE WHEN CHARINDEX(N'.' , LeafName COLLATE Latin1_General_BIN) > 0
THEN RIGHT(LeafName , CHARINDEX(N'.' , REVERSE(LeafName) COLLATE Latin1_General_BIN)-1)
ELSE N''
END,
ExtensionForFile AS
CASE WHEN Type = 0 AND CHARINDEX(N'.' , LeafName COLLATE Latin1_General_BIN) > 0
THEN RIGHT(LeafName , CHARINDEX(N'.' , REVERSE(LeafName) COLLATE Latin1_General_BIN)-1)
ELSE N''
END,
ItemChildCount int NOT NULL DEFAULT 0,
FolderChildCount int NOT NULL DEFAULT 0,
FileFormatMetaInfo varbinary(max) NULL,
FileFormatMetaInfoSize int NOT NULL DEFAULT 0,
FFMConsistent bit NULL,
ContentVersion int NOT NULL DEFAULT 0,
ListSchemaVersion int NULL,
Cl int varbinary(16) NULL,
NextBSN bigint NULL,
StreamSchema tinyint NULL);

**Id:** The **document identifier** of the document.
**SiteId**: The **Site Collection Identifier** (section 2.2.1.9) of the site collection containing the document.

**DirName**: The directory name of the document location.

**LeafName**: The leaf name of the document location.

**WebId**: The **Site Identifier** (section 2.2.1.11) of the site containing the document.

**ListId**: The **List Identifier** (section 2.2.1.5) of the list containing the document, if any, or NULL if the document is not contained in a list.

**DoclibRowId**: The row identifier for the document within the containing document library or list, if applicable.

**Type**: An integer identifier specifying the document's **Document Store Type** (section 2.2.2.4).

**SortBehavior**: An integer value specifying how an item is sorted within a view. If this parameter is set to 1, the item MUST be sorted like a folder. If this parameter is set to 0, it MUST be sorted like a file. If the document is a site (2), the value MUST be 2. Other values are invalid.

**Size**: The size of the document stream in bytes. This parameter can be set to NULL or to zero for items such as sites, folders, document libraries, and lists.

**ETagVersion**: A counter used for conflict detection that is incremented any time a change is made to this document. This takes into account the pending version update in **BumpVersion**. This is a version number separate from the user-visible versioning system in **UIVersion** and **UIVersionString**.

**EffectiveVersion**: A counter incremented any time a change is made to the document and is only used internally. This takes into account the pending version update in **BumpVersion**.

**InternalVersion**: A counter incremented any time a change is made to the document and is only used internally to join to other tables. This does not take into account the pending version update in **BumpVersion**.

**BumpVersion**: A counter incremented any time the **InternalVersion** of a document needs to be updated. It stores the pending version update to the document.

**UIVersion**: A **UI version** number associated with the document. **UIVersion** defaults to 512, which corresponds to a displayed version of 1.0.

**Dirty**: Set to one if the document has dependencies that require further processing.

**ListDataDirty**: Set to one if the document requires subsequent fix-up of link information.

**CacheParseId**: An implementation-specific identifier for an internal dependency update process. Used to manage bulk updates to documents when processed for dependency fix-up.

**DocFlags**: A **Doc Flags** value (section 2.2.2.3) specifying information about the document. See the **Doc Flags** section for details.

**ThicketFlag**: If set to one, this indicates that the document is an auxiliary thicket file, one of a set of files supporting a thicket.

** CharSet**: An optional character set associated with the document. Any **Windows code page** identifier is valid for CharSet. This value can be NULL, indicating that no Windows code page is specified for the document.

**ProgId**: An optional preferred application to open this document. The **ProgId** is used to distinguish between different applications that save files with a given file extension (that is, different editors for HTML or XML files).
**TimeCreated:** A time stamp in **UTC** format specifying when this document was created.

**TimeLastModified:** A time stamp in UTC format. The value specifies when the document was last saved. This corresponds to the actual time when the document was last modified.

**NextToLastTimeModified:** The value of **TimeLastModified** from the previous time that the document was saved.

**MetaInfoTimeLastModified:** A time stamp in UTC format specifying when the metadata information for the document was last changed.

**TimeLastWritten:** A time stamp in UTC format specifying when any changes were made to the document stream. This does not reflect changes made strictly to the **MetaInfo** or other item properties.

**DeleteTransactionId:** The transaction identifier for the implementation-specific deleted items recycle bin. A value of 0x specifies that this document is not marked as deleted. In the **Docs View** (section 2.2.6.3), **DeleteTransactionId** MUST equal 0x, because the **Docs View** only displays items that are not marked as deleted.

**SetupPathVersion:** For an **uncustomized** document, this parameter governs the **SetupPath** fragment relative to the setup path location. The following values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>This is not an uncustomized document.</td>
</tr>
<tr>
<td>2</td>
<td>The <strong>SetupPath</strong> is relative to the install location of Windows SharePoint Services 2.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).</td>
</tr>
<tr>
<td>3</td>
<td>The <strong>SetupPath</strong> is relative to the install location of Windows SharePoint Services 3.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).</td>
</tr>
<tr>
<td>4</td>
<td>The <strong>SetupPath</strong> is relative to the install location of Microsoft SharePoint Foundation 2010 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).</td>
</tr>
<tr>
<td>15</td>
<td>The <strong>SetupPath</strong> is relative to the install location of Microsoft SharePoint Foundation 2013 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\15).</td>
</tr>
<tr>
<td>16</td>
<td>The <strong>SetupPath</strong> is relative to the install location of Microsoft SharePoint Server 2016 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\16).</td>
</tr>
</tbody>
</table>

**SetupPath:** For a document that is now or once was uncustomized, this contains the setup path fragment relative to the base setup path specified by the **SetupPathVersion** value, where the content **stream** of this document can be found. This value MUST be NULL if the document was never uncustomized.

**SetupPathUser:** If this document is now or once was uncustomized, this contains the login name of the user who created the uncustomized document. This value is undefined for documents that were never uncustomized.

**CheckoutUserId:** If the document is checked out, this parameter contains the **User Identifier** (section 2.2.1.1.13) of the user who has the document checked out. Otherwise, this parameter is NULL.

**CheckoutDate:** A time stamp in UTC format indicating when this document was checked out.

**CheckoutExpires:** A time stamp in UTC format indicating when the short-term lock for this document will expire.
**VersionCreatedSinceSTCheckout:** If this parameter is one, the document version has been incremented since the document last had a short-term lock established. This is used to prevent more than one new version of the document from being created while a short-term lock is established.

**LTCheckoutUserId:** If the document is currently checked out, this parameter is a calculated column containing the value of CheckoutUserId. Otherwise, this parameter is NULL.

**VirusVendorID:** The identifier of the virus scanner that processed this document. This value MUST be NULL if this document has not been processed by a virus scanner.

**VirusStatus:** An enumerated type specifying the current virus check status of this document. This value MUST be NULL if the document has not been processed by a virus scanner. See Virus Status (section 2.2.1.2.17) for a list of valid values.

**VirusInfo:** A string containing a provider-specific message returned by the virus scanner when it last processed the document. This value MUST be NULL if the document has not been processed by a virus scanner.

**VirusInfoEx:** A binary value that specifies extensible virus information for this document, which is opaque to the back-end server.

**MetaInfo:** A metadict for the document. The metadict format is specified in the [MS-FPSE] section 2.2.2.2.11.

**MetaInfoSize:** The size, in bytes, of the document metadata info.

**MetaInfoVersion:** An integer value that tracks the version of the document metadata info.

**UnVersionedMetaInfo:** A metadict holding all version-independent metadata for the document.

**UnVersionedMetaInfoSize:** The size in bytes of UnVersionedMetaInfo.

**UnVersionedMetaInfoVersion:** An integer value that tracks the version of the UnVersionedMetaInfo metadata.

**WelcomePageUrl:** If the document is a folder, this optionally specifies a page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself and MUST be subsumed by that folder. Attempts to break out of the folder, such as "../../default.aspx", are not valid.

**WelcomePageParameters:** This parameter MUST contain any URL parameters configured for the welcome page. This value contains a query string starting with "?" or a hash parameter starting with ";#".

**IsCurrentVersion:** If this value is set to one, this row is for a current version of the document (out of all the rows corresponding to the given DirName, LeafName, each with a different Publishing Level Type value (section 2.2.2.6)). Otherwise, this row is for a historical version of the document.

**Level:** A Publishing Level Type value specifying the publishing level of this version of the document as checked out, draft, or published.

**CheckinComment:** An optional comment string associated with the document when it was last checked in or published.

**AuditFlags:** An Audit Flags value (section 2.2.2.1) determining the operations to be tracked on this document.

**InheritAuditFlags:** An Audit Flags value determining the operations to be tracked on this document's children.
DraftOwnerId: If the current publishing level of the document is a draft, this is the User Identifier (section 2.2.1.1.13) of the user who has that draft version checked out. Otherwise, this MUST be set to NULL.

UIVersionString: A calculated column presenting the value of the UIVersion column as a displayed version string, in the following format: major version value, followed by '.', followed by the minor version value. For example, UIVersion values 512, 513, 514, and 1024 correspond to UIVersionString values of 1.0, 1.1, 1.2, and 2.0, respectively.

ParentId: The document identifier of the document's parent container. For example, the document identifier of the folder or document library containing this subfolder or document. This value MUST NOT be NULL, because every item but one has a parent container. A special empty document identifier, '00000000-0000-0000-0000-000000000000', marks the parent container of the topmost item (the root site) in the site collection.

HasStream: A calculated bit indicating whether the document has an associated document stream. This MUST be set to "1" if:

- The SetupPath is NULL and the document's Document Store Type is zero (file) and it is allowed to contain a stream.
- The SetupPath is not NULL and the document content is customized.

If neither condition applies, the value MUST be "0".

ScopeId: The Scope Identifier (section 2.2.1.1.8) of the scope of the document.

BuildDependencySet: A binary array holding implementation-specific information about dependent documents. NULL indicates that there is no information about dependencies. A BuildDependencySet of size zero indicates a document with no dependencies.

ParentVersion: If the document is a transformed version of another document, this is the UIVersion value from the original document. This value MUST be NULL if the document is not the product of a document transformation.

ParentVersionString: A calculated column presenting the value of the ParentVersion column as a displayed version string. See UIVersionString for displayed version format and examples.

TransformerId: If the document is a transformed version of another document, this MUST be the GUID of the agent that performed the transformation. Otherwise, TransformerId MUST be NULL.

ParentLeafName: If the document is a transformed version of another document and the original document is in the same folder as the transformed document, this is the leaf name of the original document. This value MUST be NULL if the document is not the product of a document transformation.

IsCheckoutToLocal: A calculated bit indicating whether the document is checked out to a client's local disk. This parameter MUST be set to "1" if the DocFlags bit array includes 512 (long-term check-out to local); otherwise, it MUST be set to "0".

CtoOffset: A content type order offset. This is an implementation-specific value used by list and document library views to allow custom ordering of the menu items under the 'New' button. This value can be NULL.

Extension: A calculated field holding the file extension part, if applicable, from the LeafName value. If the LeafName value contains a "." character, Extension MUST hold the string following the last "."; otherwise, Extension MUST be set to the empty string. For example, if the full leaf name is 'document.docx.bac', Extension MUST be "bac".

ExtensionForFile: A calculated field holding the file extension part, if applicable, from the LeafName value. If the document's Document Store Type is set to "0", indicating that it is a file type,
ExtensionForFile MUST be identical to Extension; otherwise, the ExtensionForFile value MUST be an empty string.

**ItemChildCount:** The number of nonfolder children of this document.

**FolderChildCount:** The number of folder children of this document.

**FileFormatMetaInfo:** A metadict holding the file format metadata info of this document.

**FileFormatMetaInfoSize:** The size, in bytes, of the file format metadata info.

**FFMConsistent:** A bit flag specifying whether the file format metadata info is in a consistent state.

**ContentVersion:** A counter used for internal conflict detection that is incremented any time a change is made to the binary contents of this document.

**ListSchemaVersion:** A counter which keeps track of the schema version of the list containing this document for a document that can be parsed.

**ClientId:** An identifier assigned by an external client to keep track of this document.

**NextBSN:** The current BLOB sequence number for the document, used for assigning BLOB sequence numbers to stream binary pieces belonging to this document.

**StreamSchema:** The stream schema of the stream belonging to this document, if it has one indicated by HasStream.

### 2.2.6.2 AllUserData Table

The AllUserData table stores data for all list (1) and document library list items. The table provides a fixed number of generic columns in various data types, affording storage for application-defined variable schemas.

A list item can be represented by more than one row in this table, if its list's (1) schema requires more entries of a particular data type than are available in a single row. Application-defined metadata for documents in document libraries also resides in AllUserData, and is accessed by means of joins with the Docs View (section 2.2.6.3).

The AllUserData table is defined using T-SQL syntax, as follows.

```sql
TABLE AllUserData (  
    tp_Id                          int                NOT NULL,  
    tp_ListId                      uniqueidentifier   NOT NULL,  
    tp_SiteId                      uniqueidentifier NOT NULL,  
    tp_RowOrdinal                  tinyint            NOT NULL   DEFAULT ((0)),  
    tp_Version                     int                NOT NULL,  
    tp_Author                      int                NULL,  
    tp_Editor                      int                NULL,  
    tp_Modified                    datetime           NULL,  
    tp_Created                     datetime           NULL,  
    tp_Ordering                    varchar(512)       NULL,  
    tp_ThreadIndex                 varbinary(512)     NULL,  
    tp_HasAttachment               bit                NOT NULL   DEFAULT ((0)),  
    tp_ModerationStatus            int                NOT NULL   DEFAULT ((0)),  
    tp_IsCurrent                   bit                NOT NULL   DEFAULT ((1)),  
    tp_ItemOrder                   float              NULL,  
    tp_InstanceID                  int                NULL,  
    tp_GUID                        uniqueidentifier   NOT NULL   DEFAULT (newid()),  
    tp_CopySource                  nvarchar(260)      NULL       DEFAULT (NULL),  
    tp_HasCopyDestinations         bit                NULL       DEFAULT ((0)),  
    tp_AuditFlags                  int                NULL,  
    tp_InheritAuditFlags          int                NULL,  
    tp_Size                        int                NOT NULL   DEFAULT ((0)),
)
```
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp_WorkflowVersion</td>
<td>int</td>
<td>NULL,</td>
</tr>
<tr>
<td>tp_WorkflowInstanceId</td>
<td>uniqueidentifier</td>
<td>NULL,</td>
</tr>
<tr>
<td>tp_ParentId</td>
<td>uniqueidentifier</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>tp_DocId</td>
<td>uniqueidentifier</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>tp_DeleteTransactionId</td>
<td>varbinary(16)</td>
<td>NOT NULL DEFAULT 0x,</td>
</tr>
<tr>
<td>tp_ContextType</td>
<td>tinyint</td>
<td>NOT NULL DEFAULT (1),</td>
</tr>
<tr>
<td>tp_IsCurrentVersion</td>
<td>bit</td>
<td>NOT NULL DEFAULT (CONVERT ([bit],(1),0)),</td>
</tr>
<tr>
<td>tp_UIVersion</td>
<td>int</td>
<td>NOT NULL CONSTRAINT</td>
</tr>
<tr>
<td>tp_CalculatedVersion</td>
<td>int</td>
<td>NOT NULL CONSTRAINT</td>
</tr>
<tr>
<td>tp_UIVersionString</td>
<td>AS</td>
<td>(CONVERT([nvarchar],[tp_UIVersion]/(512),0)+'.'), + CONVERT([nvarchar],[tp_UIVersion]%512,0),</td>
</tr>
<tr>
<td>tp_DraftOwnerId</td>
<td>int</td>
<td>NULL DEFAULT (NULL),</td>
</tr>
<tr>
<td>tp_CheckoutUserId</td>
<td>int</td>
<td>NULL DEFAULT (NULL),</td>
</tr>
<tr>
<td>tp_AppAuthor</td>
<td>int</td>
<td>NULL,</td>
</tr>
<tr>
<td>tp_AppEditor</td>
<td>int</td>
<td>NULL,</td>
</tr>
<tr>
<td>bit1</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit2</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit3</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit4</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit5</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit6</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit7</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit8</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit9</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit10</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit11</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit12</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit13</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit14</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit15</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit16</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit17</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit18</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit19</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit20</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit21</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit22</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit23</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit24</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit25</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit26</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit27</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit28</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit29</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit30</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit31</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit32</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit33</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit34</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit35</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit36</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit37</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit38</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit39</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit40</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit41</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit42</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit43</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit44</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit45</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit46</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit47</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit48</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit49</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
</tbody>
</table>

[MS-WSSFO3] - v20170620
Windows SharePoint Services (WSS): File Operations Database Communications Version 3 Protocol
Copyright © 2017 Microsoft Corporation
Release: June 20, 2017
<table>
<thead>
<tr>
<th>Bit</th>
<th>Type</th>
<th>Sparse</th>
<th>NULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit50</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit51</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit52</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit53</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit54</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit55</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit56</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit57</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit58</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit59</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit60</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit61</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit62</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit63</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit64</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit65</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit66</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit67</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit68</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit69</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit70</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit71</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit72</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit73</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit74</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit75</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit76</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit77</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit78</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit79</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit80</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit81</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit82</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit83</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit84</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit85</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit86</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit87</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit88</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit89</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit90</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit91</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit92</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit93</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit94</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit95</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit96</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit97</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit98</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit99</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit100</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit101</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit102</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit103</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit104</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit105</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit106</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit107</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit108</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit109</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit110</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit111</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit112</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit113</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit114</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit115</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit116</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit117</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>bit118</td>
<td>bit</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
</tbody>
</table>
bit119 bit SPARSE NULL,
bit120 bit SPARSE NULL,
bit121 bit SPARSE NULL,
bit122 bit SPARSE NULL,
bit123 bit SPARSE NULL,
bit124 bit SPARSE NULL,
bit125 bit SPARSE NULL,
bit126 bit SPARSE NULL,
bit127 bit SPARSE NULL,
bit128 bit SPARSE NULL,
bit129 bit SPARSE NULL,
bit130 bit SPARSE NULL,
bit131 bit SPARSE NULL,
bit132 bit SPARSE NULL,
bit133 bit SPARSE NULL,
bit134 bit SPARSE NULL,
bit135 bit SPARSE NULL,
bit136 bit SPARSE NULL,
bit137 bit SPARSE NULL,
bit138 bit SPARSE NULL,
bit139 bit SPARSE NULL,
bit140 bit SPARSE NULL,
bit141 bit SPARSE NULL,
bit142 bit SPARSE NULL,
bit143 bit SPARSE NULL,
bit144 bit SPARSE NULL,
bit145 bit SPARSE NULL,
bit146 bit SPARSE NULL,
bit147 bit SPARSE NULL,
bit148 bit SPARSE NULL,
bit149 bit SPARSE NULL,
bit150 bit SPARSE NULL,
bit151 bit SPARSE NULL,
bit152 bit SPARSE NULL,
bit153 bit SPARSE NULL,
bit154 bit SPARSE NULL,
bit155 bit SPARSE NULL,
bit156 bit SPARSE NULL,
bit157 bit SPARSE NULL,
bit158 bit SPARSE NULL,
bit159 bit SPARSE NULL,
bit160 bit SPARSE NULL,
bit161 bit SPARSE NULL,
bit162 bit SPARSE NULL,
bit163 bit SPARSE NULL,
bit164 bit SPARSE NULL,
bit165 bit SPARSE NULL,
bit166 bit SPARSE NULL,
bit167 bit SPARSE NULL,
bit168 bit SPARSE NULL,
bit169 bit SPARSE NULL,
bit170 bit SPARSE NULL,
bit171 bit SPARSE NULL,
bit172 bit SPARSE NULL,
bit173 bit SPARSE NULL,
bit174 bit SPARSE NULL,
bit175 bit SPARSE NULL,
bit176 bit SPARSE NULL,
bit177 bit SPARSE NULL,
bit178 bit SPARSE NULL,
bit179 bit SPARSE NULL,
bit180 bit SPARSE NULL,
bit181 bit SPARSE NULL,
bit182 bit SPARSE NULL,
bit183 bit SPARSE NULL,
bit184 bit SPARSE NULL,
bit185 bit SPARSE NULL,
bit186 bit SPARSE NULL,
<table>
<thead>
<tr>
<th>bit</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit188</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit189</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit190</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit191</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit192</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit193</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit194</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit195</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit196</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit197</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit198</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit199</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit200</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit201</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit202</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit203</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit204</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit205</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit206</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit207</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit208</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit209</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit210</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit211</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit212</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit213</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit214</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit215</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit216</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit217</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit218</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit219</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit220</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit221</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit222</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit223</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit224</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit225</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit226</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit227</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit228</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit229</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit230</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit231</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit232</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit233</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit234</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit235</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit236</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit237</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit238</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit239</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit240</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit241</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit242</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit243</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit244</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit245</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit246</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit247</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit248</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit249</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit250</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit251</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit252</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit253</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit254</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit255</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit256</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit257</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit258</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit259</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit260</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit261</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit262</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit263</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit264</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit265</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit266</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit267</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit268</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit269</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit270</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit271</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit272</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit273</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit274</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit275</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit276</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit277</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit278</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit279</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit280</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit281</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit282</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit283</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit284</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit285</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit286</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit287</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit288</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit289</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit290</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit291</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit292</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit293</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit294</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit295</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit296</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit297</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit298</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit299</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit300</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit301</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit302</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit303</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit304</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit305</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit306</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit307</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit308</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit309</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit310</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit311</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit312</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit313</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit314</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit315</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit316</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit317</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit318</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit319</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit320</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit321</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit322</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit323</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit324</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit325</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 326</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 327</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 328</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 329</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 330</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 331</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 332</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 333</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 334</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 335</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 336</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 337</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 338</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 339</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 340</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 341</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 342</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 343</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 344</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 345</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 346</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 347</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 348</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 349</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 350</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 351</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 352</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 353</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 354</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 355</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 356</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 357</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 358</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 359</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 360</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 361</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 362</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 363</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 364</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 365</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 366</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 367</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 368</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 369</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 370</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 371</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 372</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 373</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 374</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 375</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 376</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 377</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 378</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 379</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 380</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 381</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 382</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 383</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 384</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 385</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 386</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 387</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 388</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 389</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 390</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 391</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 392</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 393</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit 394</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
</tbody>
</table>
bit395                  bit               SPARSE  NULL,
bit396                  bit               SPARSE  NULL,
bit397                  bit               SPARSE  NULL,
bit398                  bit               SPARSE  NULL,
bit399                  bit               SPARSE  NULL,
bit400                  bit               SPARSE  NULL,
bit401                  bit               SPARSE  NULL,
bit402                  bit               SPARSE  NULL,
bit403                  bit               SPARSE  NULL,
bit404                  bit               SPARSE  NULL,
bit405                  bit               SPARSE  NULL,
bit406                  bit               SPARSE  NULL,
bit407                  bit               SPARSE  NULL,
bit408                  bit               SPARSE  NULL,
bit409                  bit               SPARSE  NULL,
bit410                  bit               SPARSE  NULL,
bit411                  bit               SPARSE  NULL,
bit412                  bit               SPARSE  NULL,
bit413                  bit               SPARSE  NULL,
bit414                  bit               SPARSE  NULL,
bit415                  bit               SPARSE  NULL,
bit416                  bit               SPARSE  NULL,
bit417                  bit               SPARSE  NULL,
bit418                  bit               SPARSE  NULL,
bit419                  bit               SPARSE  NULL,
bit420                  bit               SPARSE  NULL,
bit421                  bit               SPARSE  NULL,
bit422                  bit               SPARSE  NULL,
bit423                  bit               SPARSE  NULL,
bit424                  bit               SPARSE  NULL,
bit425                  bit               SPARSE  NULL,
bit426                  bit               SPARSE  NULL,
bit427                  bit               SPARSE  NULL,
bit428                  bit               SPARSE  NULL,
bit429                  bit               SPARSE  NULL,
bit430                  bit               SPARSE  NULL,
bit431                  bit               SPARSE  NULL,
bit432                  bit               SPARSE  NULL,
bit433                  bit               SPARSE  NULL,
bit434                  bit               SPARSE  NULL,
bit435                  bit               SPARSE  NULL,
bit436                  bit               SPARSE  NULL,
bit437                  bit               SPARSE  NULL,
bit438                  bit               SPARSE  NULL,
bit439                  bit               SPARSE  NULL,
bit440                  bit               SPARSE  NULL,
bit441                  bit               SPARSE  NULL,
bit442                  bit               SPARSE  NULL,
bit443                  bit               SPARSE  NULL,
bit444                  bit               SPARSE  NULL,
bit445                  bit               SPARSE  NULL,
bit446                  bit               SPARSE  NULL,
bit447                  bit               SPARSE  NULL,
bit448                  bit               SPARSE  NULL,
bit449                  bit               SPARSE  NULL,
bit450                  bit               SPARSE  NULL,
bit451                  bit               SPARSE  NULL,
bit452                  bit               SPARSE  NULL,
bit453                  bit               SPARSE  NULL,
bit454                  bit               SPARSE  NULL,
bit455                  bit               SPARSE  NULL,
<table>
<thead>
<tr>
<th>bit</th>
<th>value</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit533</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit534</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit535</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit536</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit537</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit538</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit539</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit540</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit541</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit542</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit543</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit544</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit545</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit546</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit547</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit548</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit549</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit550</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit551</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit552</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit553</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit554</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit555</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit556</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit557</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit558</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit559</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit560</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit561</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit562</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit563</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit564</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit565</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit566</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit567</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit568</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit569</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit570</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit571</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit572</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit573</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit574</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit575</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit576</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit577</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit578</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit579</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit580</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit581</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit582</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit583</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit584</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit585</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit586</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit587</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit588</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit589</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit590</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit591</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit592</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit593</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit594</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit595</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit596</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit597</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit598</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit599</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit600</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>bit601</td>
<td>bit</td>
<td>SPARSE NULL,</td>
</tr>
</tbody>
</table>
bit602  bit               SPARSE  NULL,
bite603  bit               SPARSE  NULL,
bite604  bit               SPARSE  NULL,
bite605  bit               SPARSE  NULL,
bite606  bit               SPARSE  NULL,
bite607  bit               SPARSE  NULL,
bite608  bit               SPARSE  NULL,
bite609  bit               SPARSE  NULL,
bite610  bit               SPARSE  NULL,
bite611  bit               SPARSE  NULL,
bite612  bit               SPARSE  NULL,
bite613  bit               SPARSE  NULL,
bite614  bit               SPARSE  NULL,
bite615  bit               SPARSE  NULL,
bite616  bit               SPARSE  NULL,
bite617  bit               SPARSE  NULL,
bite618  bit               SPARSE  NULL,
bite619  bit               SPARSE  NULL,
bit 671  bit  SPARSE  NULL,
bit 672  bit  SPARSE  NULL,
bit 673  bit  SPARSE  NULL,
bit 674  bit  SPARSE  NULL,
bit 675  bit  SPARSE  NULL,
bit 676  bit  SPARSE  NULL,
bit 677  bit  SPARSE  NULL,
bit 678  bit  SPARSE  NULL,
bit 679  bit  SPARSE  NULL,
bit 680  bit  SPARSE  NULL,
bit 681  bit  SPARSE  NULL,
bit 682  bit  SPARSE  NULL,
bit 683  bit  SPARSE  NULL,
bit 684  bit  SPARSE  NULL,
bit 685  bit  SPARSE  NULL,
bit 686  bit  SPARSE  NULL,
bit 687  bit  SPARSE  NULL,
bit 688  bit  SPARSE  NULL,
bit 689  bit  SPARSE  NULL,
bit 690  bit  SPARSE  NULL,
bit 691  bit  SPARSE  NULL,
bit 692  bit  SPARSE  NULL,
bit 693  bit  SPARSE  NULL,
bit 694  bit  SPARSE  NULL,
bit 695  bit  SPARSE  NULL,
bit 696  bit  SPARSE  NULL,
bit 697  bit  SPARSE  NULL,
bit 698  bit  SPARSE  NULL,
bit 699  bit  SPARSE  NULL,
bit 700  bit  SPARSE  NULL,
bit 701  bit  SPARSE  NULL,
bit 702  bit  SPARSE  NULL,
bit 703  bit  SPARSE  NULL,
bit 704  bit  SPARSE  NULL,
bit 705  bit  SPARSE  NULL,
bit 706  bit  SPARSE  NULL,
bit 707  bit  SPARSE  NULL,
bit 708  bit  SPARSE  NULL,
bit 709  bit  SPARSE  NULL,
bit 710  bit  SPARSE  NULL,
bit 711  bit  SPARSE  NULL,
bit 712  bit  SPARSE  NULL,
bit 713  bit  SPARSE  NULL,
bit 714  bit  SPARSE  NULL,
bit 715  bit  SPARSE  NULL,
bit 716  bit  SPARSE  NULL,
bit 717  bit  SPARSE  NULL,
bit 718  bit  SPARSE  NULL,
bit 719  bit  SPARSE  NULL,
bit 720  bit  SPARSE  NULL,
bit 721  bit  SPARSE  NULL,
bit 722  bit  SPARSE  NULL,
bit 723  bit  SPARSE  NULL,
bit 724  bit  SPARSE  NULL,
bit 725  bit  SPARSE  NULL,
bit 726  bit  SPARSE  NULL,
bit 727  bit  SPARSE  NULL,
bit 728  bit  SPARSE  NULL,
bit 729  bit  SPARSE  NULL,
bit 730  bit  SPARSE  NULL,
<table>
<thead>
<tr>
<th>Bit</th>
<th>Value</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>740</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>741</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>742</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>743</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>744</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>745</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>746</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>747</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>748</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>749</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>750</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>751</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>752</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>753</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>754</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>755</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>756</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>757</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>758</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>759</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>760</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>761</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>762</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>763</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>764</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>765</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>766</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>767</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>768</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>769</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>770</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>771</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>772</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>773</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>774</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>775</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>776</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>777</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>778</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>779</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>780</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>781</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>782</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>783</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>784</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>785</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>786</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>787</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>788</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>789</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>790</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>791</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>792</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>793</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>794</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>795</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>796</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>797</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>798</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>799</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>801</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>802</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>803</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>804</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>805</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>806</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>807</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>808</td>
<td>bit</td>
<td>SPARSE NULL</td>
<td></td>
</tr>
<tr>
<td>bit 809</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>bit 810</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 811</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 812</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 813</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 814</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 815</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 816</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 817</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 818</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 819</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 820</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 821</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 822</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 823</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 824</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 825</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 826</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 827</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 828</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 829</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 830</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 831</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 832</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 833</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 834</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 835</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 836</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 837</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 838</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 839</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 840</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 841</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 842</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 843</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 844</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 845</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 846</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 847</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 848</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 849</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 850</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 851</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 852</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 853</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 854</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 855</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 856</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 857</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 858</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 859</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 860</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 861</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 862</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 863</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 864</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 865</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 866</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 867</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 868</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 869</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 870</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 871</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 872</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 873</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 874</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 875</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 876</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>bit 877</td>
<td>bit</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
</tbody>
</table>
bit947              bit               SPARSE  NULL,
bit948              bit               SPARSE  NULL,
bit949              bit               SPARSE  NULL,
bit950              bit               SPARSE  NULL,
bit951              bit               SPARSE  NULL,
bit952              bit               SPARSE  NULL,
bit953              bit               SPARSE  NULL,
bit954              bit               SPARSE  NULL,
bit955              bit               SPARSE  NULL,
bit956              bit               SPARSE  NULL,
bit957              bit               SPARSE  NULL,
bit958              bit               SPARSE  NULL,
bit959              bit               SPARSE  NULL,
bit960              bit               SPARSE  NULL,
bit961              bit               SPARSE  NULL,
bit962              bit               SPARSE  NULL,
bit963              bit               SPARSE  NULL,
bit964              bit               SPARSE  NULL,
bit965              bit               SPARSE  NULL,
bit966              bit               SPARSE  NULL,
bit967              bit               SPARSE  NULL,
bit968              bit               SPARSE  NULL,
bit969              bit               SPARSE  NULL,
bit970              bit               SPARSE  NULL,
bit971              bit               SPARSE  NULL,
bit972              bit               SPARSE  NULL,
bit973              bit               SPARSE  NULL,
bit974              bit               SPARSE  NULL,
bit975              bit               SPARSE  NULL,
bit976              bit               SPARSE  NULL,
bit977              bit               SPARSE  NULL,
bit978              bit               SPARSE  NULL,
bit979              bit               SPARSE  NULL,
bit980              bit               SPARSE  NULL,
bit981              bit               SPARSE  NULL,
bit982              bit               SPARSE  NULL,
bit983              bit               SPARSE  NULL,
bit984              bit               SPARSE  NULL,
bit985              bit               SPARSE  NULL,
bit986              bit               SPARSE  NULL,
bit987              bit               SPARSE  NULL,
bit988              bit               SPARSE  NULL,
bit989              bit               SPARSE  NULL,
bit990              bit               SPARSE  NULL,
bit991              bit               SPARSE  NULL,
bit992              bit               SPARSE  NULL,
bit993              bit               SPARSE  NULL,
bit994              bit               SPARSE  NULL,
bit995              bit               SPARSE  NULL,
bit996              bit               SPARSE  NULL,
bit997              bit               SPARSE  NULL,
bit998              bit               SPARSE  NULL,
bit999              bit               SPARSE  NULL,
bit1000             bit               SPARSE  NULL,
datetime1          datetime          SPARSE  NULL,
datetime2          datetime          SPARSE  NULL,
datetime3          datetime          SPARSE  NULL,
datetime4          datetime          SPARSE  NULL,
datetime5          datetime          SPARSE  NULL,
datetime6          datetime          SPARSE  NULL,
datetime7          datetime          SPARSE  NULL,
datetime8          datetime          SPARSE  NULL,
datetime9          datetime          SPARSE  NULL,
datetime10         datetime          SPARSE  NULL,
datetime11         datetime          SPARSE  NULL,
datetime12         datetime          SPARSE  NULL,
datetime13         datetime          SPARSE  NULL,
datetime14         datetime          SPARSE  NULL,
datetime15         datetime          SPARSE  NULL,
datetime85           datetime          SPARSE NULL,
datetime86           datetime          SPARSE NULL,
datetime87           datetime          SPARSE NULL,
datetime88           datetime          SPARSE NULL,
datetime89           datetime          SPARSE NULL,
datetime90           datetime          SPARSE NULL,
datetime91           datetime          SPARSE NULL,
datetime92           datetime          SPARSE NULL,
datetime93           datetime          SPARSE NULL,
datetime94           datetime          SPARSE NULL,
datetime95           datetime          SPARSE NULL,
datetime96           datetime          SPARSE NULL,
datetime97           datetime          SPARSE NULL,
datetime98           datetime          SPARSE NULL,
datetime99           datetime          SPARSE NULL,
datetime100          datetime          SPARSE NULL,
datetime101          datetime          SPARSE NULL,
datetime102          datetime          SPARSE NULL,
datetime103          datetime          SPARSE NULL,
datetime104          datetime          SPARSE NULL,
datetime105          datetime          SPARSE NULL,
datetime106          datetime          SPARSE NULL,
datetime107          datetime          SPARSE NULL,
datetime108          datetime          SPARSE NULL,
datetime109          datetime          SPARSE NULL,
datetime110          datetime          SPARSE NULL,
datetime111          datetime          SPARSE NULL,
datetime112          datetime          SPARSE NULL,
datetime113          datetime          SPARSE NULL,
datetime114          datetime          SPARSE NULL,
datetime115          datetime          SPARSE NULL,
datetime116          datetime          SPARSE NULL,
datetime117          datetime          SPARSE NULL,
datetime118          datetime          SPARSE NULL,
datetime119          datetime          SPARSE NULL,
datetime120          datetime          SPARSE NULL,
datetime121          datetime          SPARSE NULL,
datetime122          datetime          SPARSE NULL,
datetime123          datetime          SPARSE NULL,
datetime124          datetime          SPARSE NULL,
datetime125          datetime          SPARSE NULL,
datetime126          datetime          SPARSE NULL,
datetime127          datetime          SPARSE NULL,
datetime128          datetime          SPARSE NULL,
datetime129          datetime          SPARSE NULL,
datetime130          datetime          SPARSE NULL,
datetime131          datetime          SPARSE NULL,
datetime132          datetime          SPARSE NULL,
datetime133          datetime          SPARSE NULL,
datetime134          datetime          SPARSE NULL,
datetime135          datetime          SPARSE NULL,
datetime136          datetime          SPARSE NULL,
datetime137          datetime          SPARSE NULL,
datetime138          datetime          SPARSE NULL,
datetime139          datetime          SPARSE NULL,
datetime140          datetime          SPARSE NULL,
datetime141          datetime          SPARSE NULL,
datetime142          datetime          SPARSE NULL,
datetime143          datetime          SPARSE NULL,
datetime144          datetime          SPARSE NULL,
datetime145          datetime          SPARSE NULL,
datetime146          datetime          SPARSE NULL,
datetime147          datetime          SPARSE NULL,
datetime148          datetime          SPARSE NULL,
datetime149          datetime          SPARSE NULL,
datetime150          datetime          SPARSE NULL,
datetime151          datetime          SPARSE NULL,
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Type</th>
<th>Sparse</th>
<th>Null</th>
</tr>
</thead>
<tbody>
<tr>
<td>datetime223</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime224</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime225</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime226</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime227</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime228</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime229</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime230</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime231</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime232</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime233</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime234</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime235</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime236</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime237</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime238</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime239</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime240</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime241</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime242</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime243</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime244</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime245</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime246</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime247</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime248</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime249</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime250</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime251</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime252</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime253</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime254</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime255</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime256</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime257</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime258</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime259</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime260</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime261</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime262</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime263</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime264</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime265</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime266</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime267</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime268</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime269</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime270</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime271</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime272</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime273</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime274</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime275</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime276</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime277</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime278</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime279</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime280</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime281</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime282</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime283</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime284</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime285</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime286</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime287</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime288</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime289</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime290</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime291</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime292</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime293</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime294</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime295</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime296</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime297</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime298</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime299</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime300</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime301</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime302</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime303</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime304</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime305</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime306</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime307</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime308</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime309</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime310</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime311</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime312</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime313</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime314</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime315</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime316</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime317</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime318</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime319</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime320</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime321</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime322</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime323</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime324</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime325</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime326</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime327</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime328</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime329</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime330</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime331</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime332</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime333</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime334</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime335</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime336</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime337</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime338</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime339</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime340</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime341</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime342</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime343</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime344</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime345</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime346</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime347</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime348</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime349</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime350</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime351</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime352</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime353</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime354</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime355</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime356</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime357</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime358</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime359</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>datetime360</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
</tbody>
</table>
datetime361 datetime  SPARSE NULL,
datetime362 datetime  SPARSE NULL,
datetime363 datetime  SPARSE NULL,
datetime364 datetime  SPARSE NULL,
datetime365 datetime  SPARSE NULL,
datetime366 datetime  SPARSE NULL,
datetime367 datetime  SPARSE NULL,
datetime368 datetime  SPARSE NULL,
datetime369 datetime  SPARSE NULL,
datetime370 datetime  SPARSE NULL,
datetime371 datetime  SPARSE NULL,
datetime372 datetime  SPARSE NULL,
datetime373 datetime  SPARSE NULL,
datetime374 datetime  SPARSE NULL,
datetime375 datetime  SPARSE NULL,
datetime376 datetime  SPARSE NULL,
datetime377 datetime  SPARSE NULL,
datetime378 datetime  SPARSE NULL,
datetime379 datetime  SPARSE NULL,
datetime380 datetime  SPARSE NULL,
datetime381 datetime  SPARSE NULL,
datetime382 datetime  SPARSE NULL,
datetime383 datetime  SPARSE NULL,
datetime384 datetime  SPARSE NULL,
datetime385 datetime  SPARSE NULL,
datetime386 datetime  SPARSE NULL,
datetime387 datetime  SPARSE NULL,
datetime388 datetime  SPARSE NULL,
datetime389 datetime  SPARSE NULL,
datetime390 datetime  SPARSE NULL,
datetime391 datetime  SPARSE NULL,
datetime392 datetime  SPARSE NULL,
datetime393 datetime  SPARSE NULL,
datetime394 datetime  SPARSE NULL,
datetime395 datetime  SPARSE NULL,
datetime396 datetime  SPARSE NULL,
datetime397 datetime  SPARSE NULL,
datetime398 datetime  SPARSE NULL,
datetime399 datetime  SPARSE NULL,
datetime400 datetime  SPARSE NULL,
datetime401 datetime  SPARSE NULL,
datetime402 datetime  SPARSE NULL,
datetime403 datetime  SPARSE NULL,
datetime404 datetime  SPARSE NULL,
datetime405 datetime  SPARSE NULL,
datetime406 datetime  SPARSE NULL,
datetime407 datetime  SPARSE NULL,
datetime408 datetime  SPARSE NULL,
datetime409 datetime  SPARSE NULL,
datetime410 datetime  SPARSE NULL,
datetime411 datetime  SPARSE NULL,
datetime412 datetime  SPARSE NULL,
datetime413 datetime  SPARSE NULL,
datetime414 datetime  SPARSE NULL,
datetime415 datetime  SPARSE NULL,
<table>
<thead>
<tr>
<th>Datetime</th>
<th>Datetime</th>
<th>SPARSE</th>
<th>NULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>datetime430</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime431</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime432</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime433</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime434</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime435</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime436</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime437</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime438</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime439</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime440</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime441</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime442</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime443</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime444</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime445</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime446</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime447</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime448</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime449</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime450</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime451</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime452</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime453</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime454</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime455</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime456</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime457</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime458</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime459</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime460</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime461</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime462</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime463</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime464</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime465</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime466</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime467</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime468</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime469</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime470</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime471</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime472</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime473</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime474</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime475</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime476</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime477</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime478</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime479</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime480</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime481</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime482</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime483</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime484</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime485</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime486</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime487</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime488</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime489</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime490</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime491</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime492</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime493</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime494</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime495</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime496</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime497</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>datetime498</td>
<td>datetime</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>Column Name</td>
<td>Data Type</td>
<td>Options</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>datetime499</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime500</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime501</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime502</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime503</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime504</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime505</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime506</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime507</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime508</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime509</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime510</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime511</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime512</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime513</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime514</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime515</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime516</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime517</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime518</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime519</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime520</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime521</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime522</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime523</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime524</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime525</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime526</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime527</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime528</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime529</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime530</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime531</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime532</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime533</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime534</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime535</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime536</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime537</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime538</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime539</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime540</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime541</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime542</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime543</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime544</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime545</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime546</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime547</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime548</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime549</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>datetime550</td>
<td>datetime</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float1</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float2</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float3</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float4</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float5</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float6</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float7</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float8</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float9</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float10</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float11</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float12</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float13</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float14</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float15</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float16</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>float17</td>
<td>float</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
</tbody>
</table>
float18 float SPARSE NULL,
float19 float SPARSE NULL,
float20 float SPARSE NULL,
float21 float SPARSE NULL,
float22 float SPARSE NULL,
float23 float SPARSE NULL,
float24 float SPARSE NULL,
float25 float SPARSE NULL,
float26 float SPARSE NULL,
float27 float SPARSE NULL,
float28 float SPARSE NULL,
float29 float SPARSE NULL,
float30 float SPARSE NULL,
float31 float SPARSE NULL,
float32 float SPARSE NULL,
float33 float SPARSE NULL,
float34 float SPARSE NULL,
float35 float SPARSE NULL,
float36 float SPARSE NULL,
float37 float SPARSE NULL,
float38 float SPARSE NULL,
float39 float SPARSE NULL,
float40 float SPARSE NULL,
float41 float SPARSE NULL,
float42 float SPARSE NULL,
float43 float SPARSE NULL,
float44 float SPARSE NULL,
float45 float SPARSE NULL,
float46 float SPARSE NULL,
float47 float SPARSE NULL,
float48 float SPARSE NULL,
float49 float SPARSE NULL,
float50 float SPARSE NULL,
float51 float SPARSE NULL,
float52 float SPARSE NULL,
float53 float SPARSE NULL,
float54 float SPARSE NULL,
float55 float SPARSE NULL,
float56 float SPARSE NULL,
float57 float SPARSE NULL,
float58 float SPARSE NULL,
float59 float SPARSE NULL,
float60 float SPARSE NULL,
float61 float SPARSE NULL,
float62 float SPARSE NULL,
float63 float SPARSE NULL,
float64 float SPARSE NULL,
float65 float SPARSE NULL,
float66 float SPARSE NULL,
float67 float SPARSE NULL,
float68 float SPARSE NULL,
float69 float SPARSE NULL,
float70 float SPARSE NULL,
float71 float SPARSE NULL,
float72 float SPARSE NULL,
float73 float SPARSE NULL,
float74 float SPARSE NULL,
float75 float SPARSE NULL,
float76 float SPARSE NULL,
float77 float SPARSE NULL,
float78 float SPARSE NULL,
<p>| float87 | float | SPARSE | NULL, |
| float88 | float | SPARSE | NULL, |
| float89 | float | SPARSE | NULL, |
| float90 | float | SPARSE | NULL, |
| float91 | float | SPARSE | NULL, |
| float92 | float | SPARSE | NULL, |
| float93 | float | SPARSE | NULL, |
| float94 | float | SPARSE | NULL, |
| float95 | float | SPARSE | NULL, |
| float96 | float | SPARSE | NULL, |
| float97 | float | SPARSE | NULL, |
| float98 | float | SPARSE | NULL, |
| float99 | float | SPARSE | NULL, |
| float100 | float | SPARSE | NULL, |
| float101 | float | SPARSE | NULL, |
| float102 | float | SPARSE | NULL, |
| float103 | float | SPARSE | NULL, |
| float104 | float | SPARSE | NULL, |
| float105 | float | SPARSE | NULL, |
| float106 | float | SPARSE | NULL, |
| float107 | float | SPARSE | NULL, |
| float108 | float | SPARSE | NULL, |
| float109 | float | SPARSE | NULL, |
| float110 | float | SPARSE | NULL, |
| float111 | float | SPARSE | NULL, |
| float112 | float | SPARSE | NULL, |
| float113 | float | SPARSE | NULL, |
| float114 | float | SPARSE | NULL, |
| float115 | float | SPARSE | NULL, |
| float116 | float | SPARSE | NULL, |
| float117 | float | SPARSE | NULL, |
| float118 | float | SPARSE | NULL, |
| float119 | float | SPARSE | NULL, |
| float120 | float | SPARSE | NULL, |
| float121 | float | SPARSE | NULL, |
| float122 | float | SPARSE | NULL, |
| float123 | float | SPARSE | NULL, |
| float124 | float | SPARSE | NULL, |
| float125 | float | SPARSE | NULL, |
| float126 | float | SPARSE | NULL, |
| float127 | float | SPARSE | NULL, |
| float128 | float | SPARSE | NULL, |
| float129 | float | SPARSE | NULL, |
| float130 | float | SPARSE | NULL, |
| float131 | float | SPARSE | NULL, |
| float132 | float | SPARSE | NULL, |
| float133 | float | SPARSE | NULL, |
| float134 | float | SPARSE | NULL, |
| float135 | float | SPARSE | NULL, |
| float136 | float | SPARSE | NULL, |
| float137 | float | SPARSE | NULL, |
| float138 | float | SPARSE | NULL, |
| float139 | float | SPARSE | NULL, |
| float140 | float | SPARSE | NULL, |
| float141 | float | SPARSE | NULL, |
| float142 | float | SPARSE | NULL, |
| float143 | float | SPARSE | NULL, |
| float144 | float | SPARSE | NULL, |
| float145 | float | SPARSE | NULL, |
| float146 | float | SPARSE | NULL, |
| float147 | float | SPARSE | NULL, |
| float148 | float | SPARSE | NULL, |
| float149 | float | SPARSE | NULL, |
| float150 | float | SPARSE | NULL, |
| float151 | float | SPARSE | NULL, |
| float152 | float | SPARSE | NULL, |
| float153 | float | SPARSE | NULL, |
| float154 | float | SPARSE | NULL, |
| float155 | float | SPARSE | NULL, |</p>
<table>
<thead>
<tr>
<th>float156</th>
<th>float</th>
<th>SPARSE  NULL,</th>
</tr>
</thead>
<tbody>
<tr>
<td>float157</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float158</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float159</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float160</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float161</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float162</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float163</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float164</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float165</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float166</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float167</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float168</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float169</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float170</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float171</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float172</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float173</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float174</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float175</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float176</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float177</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float178</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float179</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float180</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float181</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float182</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float183</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float184</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float185</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float186</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float187</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float188</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float189</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float190</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float191</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float192</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float193</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float194</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float195</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float196</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float197</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float198</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float199</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float200</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float201</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float202</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float203</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float204</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float205</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float206</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float207</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float208</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float209</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float210</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float211</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float212</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float213</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float214</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float215</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float216</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float217</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float218</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float219</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float220</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float221</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float222</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float223</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>float224</td>
<td>float</td>
<td>SPARSE  NULL,</td>
</tr>
</tbody>
</table>
float225 float SPARSE NULL,
float226 float SPARSE NULL,
float227 float SPARSE NULL,
float228 float SPARSE NULL,
float229 float SPARSE NULL,
float230 float SPARSE NULL,
float231 float SPARSE NULL,
float232 float SPARSE NULL,
float233 float SPARSE NULL,
float234 float SPARSE NULL,
float235 float SPARSE NULL,
float236 float SPARSE NULL,
float237 float SPARSE NULL,
float238 float SPARSE NULL,
float239 float SPARSE NULL,
float240 float SPARSE NULL,
float241 float SPARSE NULL,
float242 float SPARSE NULL,
float243 float SPARSE NULL,
float244 float SPARSE NULL,
float245 float SPARSE NULL,
float246 float SPARSE NULL,
float247 float SPARSE NULL,
float248 float SPARSE NULL,
float249 float SPARSE NULL,
float250 float SPARSE NULL,
float251 float SPARSE NULL,
float252 float SPARSE NULL,
float253 float SPARSE NULL,
float254 float SPARSE NULL,
float255 float SPARSE NULL,
float256 float SPARSE NULL,
float257 float SPARSE NULL,
float258 float SPARSE NULL,
float259 float SPARSE NULL,
float260 float SPARSE NULL,
float261 float SPARSE NULL,
float262 float SPARSE NULL,
float263 float SPARSE NULL,
float264 float SPARSE NULL,
float265 float SPARSE NULL,
float266 float SPARSE NULL,
float267 float SPARSE NULL,
float268 float SPARSE NULL,
float269 float SPARSE NULL,
float270 float SPARSE NULL,
float271 float SPARSE NULL,
float272 float SPARSE NULL,
float273 float SPARSE NULL,
float274 float SPARSE NULL,
float275 float SPARSE NULL,
float276 float SPARSE NULL,
float277 float SPARSE NULL,
float278 float SPARSE NULL,
float279 float SPARSE NULL,
float280 float SPARSE NULL,
float281 float SPARSE NULL,
float282 float SPARSE NULL,
float283 float SPARSE NULL,
float284 float SPARSE NULL,
float285 float SPARSE NULL,
float286 float SPARSE NULL,
float287 float SPARSE NULL,
float288 float SPARSE NULL,
float289 float SPARSE NULL,
float290 float SPARSE NULL,
float291 float SPARSE NULL,
float292 float SPARSE NULL,
float293 float SPARSE NULL,
float294 float SPARSE NULL,
float295 float SPARSE NULL,
float296 float SPARSE NULL,
float297 float SPARSE NULL,
float298 float SPARSE NULL,
float299 float SPARSE NULL,
float300 float SPARSE NULL,
float301 float SPARSE NULL,
float302 float SPARSE NULL,
float303 float SPARSE NULL,
float304 float SPARSE NULL,
float305 float SPARSE NULL,
float306 float SPARSE NULL,
float307 float SPARSE NULL,
float308 float SPARSE NULL,
float309 float SPARSE NULL,
float310 float SPARSE NULL,
float311 float SPARSE NULL,
float312 float SPARSE NULL,
float313 float SPARSE NULL,
float314 float SPARSE NULL,
float315 float SPARSE NULL,
float316 float SPARSE NULL,
float317 float SPARSE NULL,
float318 float SPARSE NULL,
float319 float SPARSE NULL,
float320 float SPARSE NULL,
float321 float SPARSE NULL,
float322 float SPARSE NULL,
float323 float SPARSE NULL,
float324 float SPARSE NULL,
float325 float SPARSE NULL,
float326 float SPARSE NULL,
float327 float SPARSE NULL,
float328 float SPARSE NULL,
float329 float SPARSE NULL,
float330 float SPARSE NULL,
float331 float SPARSE NULL,
float332 float SPARSE NULL,
float333 float SPARSE NULL,
float334 float SPARSE NULL,
float335 float SPARSE NULL,
float336 float SPARSE NULL,
float337 float SPARSE NULL,
float338 float SPARSE NULL,
float339 float SPARSE NULL,
float340 float SPARSE NULL,
float341 float SPARSE NULL,
float342 float SPARSE NULL,
float343 float SPARSE NULL,
float344 float SPARSE NULL,
float345 float SPARSE NULL,
float346 float SPARSE NULL,
float347 float SPARSE NULL,
float348 float SPARSE NULL,
float349 float SPARSE NULL,
float350 float SPARSE NULL,
float351 float SPARSE NULL,
float352 float SPARSE NULL,
float353 float SPARSE NULL,
float354 float SPARSE NULL,
float355 float SPARSE NULL,
float356 float SPARSE NULL,
float357 float SPARSE NULL,
float358 float SPARSE NULL,
float359 float SPARSE NULL,
float360 float SPARSE NULL,
float361 float SPARSE NULL,
float362 float SPARSE NULL,
float363 float SPARSE NULL,
float364 float SPARSE NULL,
float365 float SPARSE NULL,
float366 float SPARSE NULL,
float367 float SPARSE NULL,
float368 float SPARSE NULL,
float369 float SPARSE NULL,
float370 float SPARSE NULL,
float371 float SPARSE NULL,
float372 float SPARSE NULL,
float373 float SPARSE NULL,
float374 float SPARSE NULL,
float375 float SPARSE NULL,
float376 float SPARSE NULL,
float377 float SPARSE NULL,
float378 float SPARSE NULL,
float379 float SPARSE NULL,
float380 float SPARSE NULL,
float381 float SPARSE NULL,
float382 float SPARSE NULL,
float383 float SPARSE NULL,
float384 float SPARSE NULL,
float385 float SPARSE NULL,
float386 float SPARSE NULL,
float387 float SPARSE NULL,
float388 float SPARSE NULL,
float389 float SPARSE NULL,
float390 float SPARSE NULL,
float391 float SPARSE NULL,
float392 float SPARSE NULL,
float393 float SPARSE NULL,
float394 float SPARSE NULL,
float395 float SPARSE NULL,
float396 float SPARSE NULL,
float397 float SPARSE NULL,
float398 float SPARSE NULL,
float399 float SPARSE NULL,
float400 float SPARSE NULL,
float401 float SPARSE NULL,
float402 float SPARSE NULL,
float403 float SPARSE NULL,
float404 float SPARSE NULL,
float405 float SPARSE NULL,
float406 float SPARSE NULL,
float407 float SPARSE NULL,
float408 float SPARSE NULL,
float409 float SPARSE NULL,
float410 float SPARSE NULL,
float411 float SPARSE NULL,
float412 float SPARSE NULL,
float413 float SPARSE NULL,
float414 float SPARSE NULL,
float415 float SPARSE NULL,
float416 float SPARSE NULL,
float417 float SPARSE NULL,
float418 float SPARSE NULL,
float419 float SPARSE NULL,
float420 float SPARSE NULL,
float421 float SPARSE NULL,
float422 float SPARSE NULL,
float423 float SPARSE NULL,
float424 float SPARSE NULL,
float425 float SPARSE NULL,
float426 float SPARSE NULL,
float427 float SPARSE NULL,
float428 float SPARSE NULL,
float429 float SPARSE NULL,
float430 float SPARSE NULL,
float431 float SPARSE NULL,
float432 float SPARSE NULL,
float433 float SPARSE NULL,
float434 float SPARSE NULL,
float435 float SPARSE NULL,
float436 float SPARSE NULL,
float437 float SPARSE NULL,
float438 float SPARSE NULL,
float439 float SPARSE NULL,
float440 float SPARSE NULL,
float441 float SPARSE NULL,
float442 float SPARSE NULL,
float443 float SPARSE NULL,
float444 float SPARSE NULL,
float445 float SPARSE NULL,
float446 float SPARSE NULL,
float447 float SPARSE NULL,
float448 float SPARSE NULL,
float449 float SPARSE NULL,
float450 float SPARSE NULL,
float451 float SPARSE NULL,
float452 float SPARSE NULL,
float453 float SPARSE NULL,
float454 float SPARSE NULL,
float455 float SPARSE NULL,
float456 float SPARSE NULL,
float457 float SPARSE NULL,
float458 float SPARSE NULL,
float459 float SPARSE NULL,
float460 float SPARSE NULL,
float461 float SPARSE NULL,
float462 float SPARSE NULL,
float463 float SPARSE NULL,
float464 float SPARSE NULL,
float465 float SPARSE NULL,
float466 float SPARSE NULL,
float467 float SPARSE NULL,
float468 float SPARSE NULL,
float469 float SPARSE NULL,
float470 float SPARSE NULL,
float471 float SPARSE NULL,
float472 float SPARSE NULL,
float473 float SPARSE NULL,
float474 float SPARSE NULL,
float475 float SPARSE NULL,
float476 float SPARSE NULL,
float477 float SPARSE NULL,
float478 float SPARSE NULL,
float479 float SPARSE NULL,
float480 float SPARSE NULL,
float481 float SPARSE NULL,
float482 float SPARSE NULL,
float483 float SPARSE NULL,
float501 float SPARSE NULL,
float502 float SPARSE NULL,
float503 float SPARSE NULL,
float504 float SPARSE NULL,
float505 float SPARSE NULL,
float506 float SPARSE NULL,
float507 float SPARSE NULL,
float508 float SPARSE NULL,
float509 float SPARSE NULL,
float510 float SPARSE NULL,
float511 float SPARSE NULL,
float512 float SPARSE NULL,
float513 float SPARSE NULL,
float514 float SPARSE NULL,
float515 float SPARSE NULL,
float516 float SPARSE NULL,
float517 float SPARSE NULL,
float518 float SPARSE NULL,
float519 float SPARSE NULL,
float520 float SPARSE NULL,
float521 float SPARSE NULL,
float522 float SPARSE NULL,
float523 float SPARSE NULL,
float524 float SPARSE NULL,
float525 float SPARSE NULL,
float526 float SPARSE NULL,
float527 float SPARSE NULL,
float528 float SPARSE NULL,
float529 float SPARSE NULL,
float530 float SPARSE NULL,
float531 float SPARSE NULL,
float532 float SPARSE NULL,
float533 float SPARSE NULL,
float534 float SPARSE NULL,
float535 float SPARSE NULL,
float536 float SPARSE NULL,
float537 float SPARSE NULL,
float538 float SPARSE NULL,
float539 float SPARSE NULL,
float540 float SPARSE NULL,
float541 float SPARSE NULL,
float542 float SPARSE NULL,
float543 float SPARSE NULL,
float544 float SPARSE NULL,
float545 float SPARSE NULL,
float546 float SPARSE NULL,
float547 float SPARSE NULL,
float548 float SPARSE NULL,
float549 float SPARSE NULL,
float550 float SPARSE NULL,
int1 int SPARSE NULL,
int2 int SPARSE NULL,
int3 int SPARSE NULL,
int4 int SPARSE NULL,
int5 int SPARSE NULL,
int6 int SPARSE NULL,
int7 int SPARSE NULL,
int8 int SPARSE NULL,
int9 int SPARSE NULL,
int10 int SPARSE NULL,
int11 int SPARSE NULL,
int12 int SPARSE NULL,
int13 int SPARSE NULL,
int14 int SPARSE NULL,
int15 int SPARSE NULL,
int16 int SPARSE NULL,
int17 int SPARSE NULL,
int18 int SPARSE NULL,
int19 int SPARSE NULL,
int20 int SPARSE NULL,
int21 int SPARSE NULL,
int22 int SPARSE NULL,
int23 int SPARSE NULL,
int24 int SPARSE NULL,
int25 int SPARSE NULL,
int26 int SPARSE NULL,
int27 int SPARSE NULL,
int28 int SPARSE NULL,
int29 int SPARSE NULL,
int30 int SPARSE NULL,
int31 int SPARSE NULL,
int32 int SPARSE NULL,
int33 int SPARSE NULL,
int34 int SPARSE NULL,
int35 int SPARSE NULL,
int36 int SPARSE NULL,
int37 int SPARSE NULL,
int38 int SPARSE NULL,
int39 int SPARSE NULL,
int40 int SPARSE NULL,
int41 int SPARSE NULL,
int42 int SPARSE NULL,
int43 int SPARSE NULL,
int44 int SPARSE NULL,
int45 int SPARSE NULL,
int46 int SPARSE NULL,
int47 int SPARSE NULL,
int48 int SPARSE NULL,
int49 int SPARSE NULL,
int50 int SPARSE NULL,
int51 int SPARSE NULL,
int52 int SPARSE NULL,
int53 int SPARSE NULL,
int54 int SPARSE NULL,
int55 int SPARSE NULL,
int56 int SPARSE NULL,
int57 int SPARSE NULL,
int58 int SPARSE NULL,
int59 int SPARSE NULL,
int60 int SPARSE NULL,
int61 int SPARSE NULL,
int62 int SPARSE NULL,
int63 int SPARSE NULL,
int64 int SPARSE NULL,
int65 int SPARSE NULL,
int66 int SPARSE NULL,
int67 int SPARSE NULL,
int68 int SPARSE NULL,
int69 int SPARSE NULL,
int70 int SPARSE NULL,
int71 int SPARSE NULL,
int72 int SPARSE NULL,
int73 int SPARSE NULL,
int74 int SPARSE NULL,
int75 int SPARSE NULL,
int76 int SPARSE NULL,
int77 int SPARSE NULL,
int78 int SPARSE NULL,
int79 int SPARSE NULL,
int80 int SPARSE NULL,
int81 int SPARSE NULL,
int82 int SPARSE NULL,
int83 int SPARSE NULL,
int84 int SPARSE NULL,
int85 int SPARSE NULL,
int86 int SPARSE NULL,
int87 int SPARSE NULL,
int88 int SPARSE NULL,
int89                int               SPARSE  NULL,
int90                int               SPARSE  NULL,
int91                int               SPARSE  NULL,
int92                int               SPARSE  NULL,
int93                int               SPARSE  NULL,
int94                int               SPARSE  NULL,
int95                int               SPARSE  NULL,
int96                int               SPARSE  NULL,
int97                int               SPARSE  NULL,
int98                int               SPARSE  NULL,
int99                int               SPARSE  NULL,
int100               int               SPARSE  NULL,
int101               int               SPARSE  NULL,
int102               int               SPARSE  NULL,
int103               int               SPARSE  NULL,
int104               int               SPARSE  NULL,
int105               int               SPARSE  NULL,
int106               int               SPARSE  NULL,
int107               int               SPARSE  NULL,
int108               int               SPARSE  NULL,
int109               int               SPARSE  NULL,
int110               int               SPARSE  NULL,
int111               int               SPARSE  NULL,
int112               int               SPARSE  NULL,
int113               int               SPARSE  NULL,
int114               int               SPARSE  NULL,
int115               int               SPARSE  NULL,
int116               int               SPARSE  NULL,
int117               int               SPARSE  NULL,
int118               int               SPARSE  NULL,
int119               int               SPARSE  NULL,
int120               int               SPARSE  NULL,
int121               int               SPARSE  NULL,
int122               int               SPARSE  NULL,
int123               int               SPARSE  NULL,
int124               int               SPARSE  NULL,
int125               int               SPARSE  NULL,
int126               int               SPARSE  NULL,
int127               int               SPARSE  NULL,
int128               int               SPARSE  NULL,
int129               int               SPARSE  NULL,
int130               int               SPARSE  NULL,
int131               int               SPARSE  NULL,
int132               int               SPARSE  NULL,
int133               int               SPARSE  NULL,
int134               int               SPARSE  NULL,
int135               int               SPARSE  NULL,
int136               int               SPARSE  NULL,
int137               int               SPARSE  NULL,
int138               int               SPARSE  NULL,
int139               int               SPARSE  NULL,
int140               int               SPARSE  NULL,
int141               int               SPARSE  NULL,
int142               int               SPARSE  NULL,
int143               int               SPARSE  NULL,
int144               int               SPARSE  NULL,
int145               int               SPARSE  NULL,
int146               int               SPARSE  NULL,
int147               int               SPARSE  NULL,
int148               int               SPARSE  NULL,
int149               int               SPARSE  NULL,
int150               int               SPARSE  NULL,
int151               int               SPARSE  NULL,
int152               int               SPARSE  NULL,
int153               int               SPARSE  NULL,
int154               int               SPARSE  NULL,
int155               int               SPARSE  NULL,
int156               int               SPARSE  NULL,
int157               int               SPARSE  NULL,
int158                  int               SPARSE  NULL,
int159                  int               SPARSE  NULL,
int160                  int               SPARSE  NULL,
int161                  int               SPARSE  NULL,
int162                  int               SPARSE  NULL,
int163                  int               SPARSE  NULL,
int164                  int               SPARSE  NULL,
int165                  int               SPARSE  NULL,
int166                  int               SPARSE  NULL,
int167                  int               SPARSE  NULL,
int168                  int               SPARSE  NULL,
int169                  int               SPARSE  NULL,
int170                  int               SPARSE  NULL,
int171                  int               SPARSE  NULL,
int172                  int               SPARSE  NULL,
int173                  int               SPARSE  NULL,
int174                  int               SPARSE  NULL,
int175                  int               SPARSE  NULL,
int176                  int               SPARSE  NULL,
int177                  int               SPARSE  NULL,
int178                  int               SPARSE  NULL,
int179                  int               SPARSE  NULL,
int180                  int               SPARSE  NULL,
int181                  int               SPARSE  NULL,
int182                  int               SPARSE  NULL,
int183                  int               SPARSE  NULL,
int184                  int               SPARSE  NULL,
int185                  int               SPARSE  NULL,
int186                  int               SPARSE  NULL,
int187                  int               SPARSE  NULL,
int188                  int               SPARSE  NULL,
int189                  int               SPARSE  NULL,
int190                  int               SPARSE  NULL,
int191                  int               SPARSE  NULL,
int192                  int               SPARSE  NULL,
int193                  int               SPARSE  NULL,
int194                  int               SPARSE  NULL,
int195                  int               SPARSE  NULL,
int196                  int               SPARSE  NULL,
int197                  int               SPARSE  NULL,
int198                  int               SPARSE  NULL,
int199                  int               SPARSE  NULL,
int200                  int               SPARSE  NULL,
int201                  int               SPARSE  NULL,
int202                  int               SPARSE  NULL,
int203                  int               SPARSE  NULL,
int204                  int               SPARSE  NULL,
int205                  int               SPARSE  NULL,
int206                  int               SPARSE  NULL,
int207                  int               SPARSE  NULL,
int208                  int               SPARSE  NULL,
int209                  int               SPARSE  NULL,
int210                  int               SPARSE  NULL,
int211                  int               SPARSE  NULL,
int212                  int               SPARSE  NULL,
int213                  int               SPARSE  NULL,
int214                  int               SPARSE  NULL,
int215                  int               SPARSE  NULL,
int216                  int               SPARSE  NULL,
int217                  int               SPARSE  NULL,
int218                  int               SPARSE  NULL,
int219                  int               SPARSE  NULL,
int220                  int               SPARSE  NULL,
int221                  int               SPARSE  NULL,
int222                  int               SPARSE  NULL,
int223                  int               SPARSE  NULL,
int224                  int               SPARSE  NULL,
int225                  int               SPARSE  NULL,
int226                  int               SPARSE  NULL,
<p>| int227 | int | SPARSE NULL, |
| int228 | int | SPARSE NULL, |
| int229 | int | SPARSE NULL, |
| int230 | int | SPARSE NULL, |
| int231 | int | SPARSE NULL, |
| int232 | int | SPARSE NULL, |
| int233 | int | SPARSE NULL, |
| int234 | int | SPARSE NULL, |
| int235 | int | SPARSE NULL, |
| int236 | int | SPARSE NULL, |
| int237 | int | SPARSE NULL, |
| int238 | int | SPARSE NULL, |
| int239 | int | SPARSE NULL, |
| int240 | int | SPARSE NULL, |
| int241 | int | SPARSE NULL, |
| int242 | int | SPARSE NULL, |
| int243 | int | SPARSE NULL, |
| int244 | int | SPARSE NULL, |
| int245 | int | SPARSE NULL, |
| int246 | int | SPARSE NULL, |
| int247 | int | SPARSE NULL, |
| int248 | int | SPARSE NULL, |
| int249 | int | SPARSE NULL, |
| int250 | int | SPARSE NULL, |
| int251 | int | SPARSE NULL, |
| int252 | int | SPARSE NULL, |
| int253 | int | SPARSE NULL, |
| int254 | int | SPARSE NULL, |
| int255 | int | SPARSE NULL, |
| int256 | int | SPARSE NULL, |
| int257 | int | SPARSE NULL, |
| int258 | int | SPARSE NULL, |
| int259 | int | SPARSE NULL, |
| int260 | int | SPARSE NULL, |
| int261 | int | SPARSE NULL, |
| int262 | int | SPARSE NULL, |
| int263 | int | SPARSE NULL, |
| int264 | int | SPARSE NULL, |
| int265 | int | SPARSE NULL, |
| int266 | int | SPARSE NULL, |
| int267 | int | SPARSE NULL, |
| int268 | int | SPARSE NULL, |
| int269 | int | SPARSE NULL, |
| int270 | int | SPARSE NULL, |
| int271 | int | SPARSE NULL, |
| int272 | int | SPARSE NULL, |
| int273 | int | SPARSE NULL, |
| int274 | int | SPARSE NULL, |
| int275 | int | SPARSE NULL, |
| int276 | int | SPARSE NULL, |
| int277 | int | SPARSE NULL, |
| int278 | int | SPARSE NULL, |
| int279 | int | SPARSE NULL, |
| int280 | int | SPARSE NULL, |
| int281 | int | SPARSE NULL, |
| int282 | int | SPARSE NULL, |
| int283 | int | SPARSE NULL, |
| int284 | int | SPARSE NULL, |
| int285 | int | SPARSE NULL, |
| int286 | int | SPARSE NULL, |
| int287 | int | SPARSE NULL, |
| int288 | int | SPARSE NULL, |
| int289 | int | SPARSE NULL, |
| int290 | int | SPARSE NULL, |
| int291 | int | SPARSE NULL, |
| int292 | int | SPARSE NULL, |
| int293 | int | SPARSE NULL, |
| int294 | int | SPARSE NULL, |
| int295 | int | SPARSE NULL, |</p>
<table>
<thead>
<tr>
<th>int296</th>
<th>int</th>
<th>SPARSE NULL,</th>
</tr>
</thead>
<tbody>
<tr>
<td>int297</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int298</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int299</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int300</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int301</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int302</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int303</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int304</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int305</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int306</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int307</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int308</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int309</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int310</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int311</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int312</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int313</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int314</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int315</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int316</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int317</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int318</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int319</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int320</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int321</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int322</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int323</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int324</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int325</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int326</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int327</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int328</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int329</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int330</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int331</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int332</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int333</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int334</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int335</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int336</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int337</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int338</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int339</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int340</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int341</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int342</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int343</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int344</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int345</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int346</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int347</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int348</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int349</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int350</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int351</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int352</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int353</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int354</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int355</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int356</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int357</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int358</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int359</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int360</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int361</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int362</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int363</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>int364</td>
<td>int</td>
<td>SPARSE NULL,</td>
</tr>
</tbody>
</table>
int365  int       SPARSE NULL,
int366  int       SPARSE NULL,
int367  int       SPARSE NULL,
int368  int       SPARSE NULL,
int369  int       SPARSE NULL,
int370  int       SPARSE NULL,
int371  int       SPARSE NULL,
int372  int       SPARSE NULL,
int373  int       SPARSE NULL,
int374  int       SPARSE NULL,
int375  int       SPARSE NULL,
int376  int       SPARSE NULL,
int377  int       SPARSE NULL,
int378  int       SPARSE NULL,
int379  int       SPARSE NULL,
int380  int       SPARSE NULL,
int381  int       SPARSE NULL,
int382  int       SPARSE NULL,
int383  int       SPARSE NULL,
int384  int       SPARSE NULL,
int385  int       SPARSE NULL,
int386  int       SPARSE NULL,
int387  int       SPARSE NULL,
int388  int       SPARSE NULL,
int389  int       SPARSE NULL,
int390  int       SPARSE NULL,
int391  int       SPARSE NULL,
int392  int       SPARSE NULL,
int393  int       SPARSE NULL,
int394  int       SPARSE NULL,
int395  int       SPARSE NULL,
int396  int       SPARSE NULL,
int397  int       SPARSE NULL,
int398  int       SPARSE NULL,
int399  int       SPARSE NULL,
int400  int       SPARSE NULL,
int401  int       SPARSE NULL,
int402  int       SPARSE NULL,
int403  int       SPARSE NULL,
int404  int       SPARSE NULL,
int405  int       SPARSE NULL,
int406  int       SPARSE NULL,
int407  int       SPARSE NULL,
int408  int       SPARSE NULL,
int409  int       SPARSE NULL,
int410  int       SPARSE NULL,
int411  int       SPARSE NULL,
int412  int       SPARSE NULL,
int413  int       SPARSE NULL,
int414  int       SPARSE NULL,
int415  int       SPARSE NULL,
int416  int       SPARSE NULL,
int417  int       SPARSE NULL,
int418  int       SPARSE NULL,
int419  int       SPARSE NULL,
int420  int       SPARSE NULL,
int421  int       SPARSE NULL,
int422  int       SPARSE NULL,
int423  int       SPARSE NULL,
int424  int       SPARSE NULL,
int425  int       SPARSE NULL,
int426  int       SPARSE NULL,
<table>
<thead>
<tr>
<th>int434</th>
<th>int</th>
<th>SPARSE  NULL,</th>
</tr>
</thead>
<tbody>
<tr>
<td>int435</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int436</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int437</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int438</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int439</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int440</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int441</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int442</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int443</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int444</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int445</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int446</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int447</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int448</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int449</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int450</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int451</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int452</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int453</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int454</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int455</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int456</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int457</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int458</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int459</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int460</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int461</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int462</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int463</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int464</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int465</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int466</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int467</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int468</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int469</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int470</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int471</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int472</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int473</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int474</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int475</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int476</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int477</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int478</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int479</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int480</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int481</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int482</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int483</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int484</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int485</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int486</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int487</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int488</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int489</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int490</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int491</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int492</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int493</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int494</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int495</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int496</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int497</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int498</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int499</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int500</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int501</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
<tr>
<td>int502</td>
<td>int</td>
<td>SPARSE  NULL,</td>
</tr>
</tbody>
</table>
int503                  int               SPARSE  NULL,
int504                  int               SPARSE  NULL,
int505                  int               SPARSE  NULL,
int506                  int               SPARSE  NULL,
int507                  int               SPARSE  NULL,
int508                  int               SPARSE  NULL,
int509                  int               SPARSE  NULL,
int510                  int               SPARSE  NULL,
int511                  int               SPARSE  NULL,
int512                  int               SPARSE  NULL,
int513                  int               SPARSE  NULL,
int514                  int               SPARSE  NULL,
int515                  int               SPARSE  NULL,
int516                  int               SPARSE  NULL,
int517                  int               SPARSE  NULL,
int518                  int               SPARSE  NULL,
int519                  int               SPARSE  NULL,
int520                  int               SPARSE  NULL,
int521                  int               SPARSE  NULL,
int522                  int               SPARSE  NULL,
int523                  int               SPARSE  NULL,
int524                  int               SPARSE  NULL,
int525                  int               SPARSE  NULL,
int526                  int               SPARSE  NULL,
int527                  int               SPARSE  NULL,
int528                  int               SPARSE  NULL,
int529                  int               SPARSE  NULL,
int530                  int               SPARSE  NULL,
int531                  int               SPARSE  NULL,
int532                  int               SPARSE  NULL,
int533                  int               SPARSE  NULL,
int534                  int               SPARSE  NULL,
int535                  int               SPARSE  NULL,
int536                  int               SPARSE  NULL,
int537                  int               SPARSE  NULL,
int538                  int               SPARSE  NULL,
int539                  int               SPARSE  NULL,
int540                  int               SPARSE  NULL,
int541                  int               SPARSE  NULL,
int542                  int               SPARSE  NULL,
int543                  int               SPARSE  NULL,
int544                  int               SPARSE  NULL,
int545                  int               SPARSE  NULL,
int546                  int               SPARSE  NULL,
int547                  int               SPARSE  NULL,
int548                  int               SPARSE  NULL,
int549                  int               SPARSE  NULL,
int550                  int               SPARSE  NULL,
int551                  int               SPARSE  NULL,
int552                  int               SPARSE  NULL,
int553                  int               SPARSE  NULL,
int554                  int               SPARSE  NULL,
int555                  int               SPARSE  NULL,
int556                  int               SPARSE  NULL,
int557                  int               SPARSE  NULL,
int558                  int               SPARSE  NULL,
int559                  int               SPARSE  NULL,
int560                  int               SPARSE  NULL,
int561                  int               SPARSE  NULL,
int562                  int               SPARSE  NULL,
int563                  int               SPARSE  NULL,
int564                  int               SPARSE  NULL,
int565                  int               SPARSE  NULL,
int566                  int               SPARSE  NULL,
int567                  int               SPARSE  NULL,
int568                  int               SPARSE  NULL,
int569                  int               SPARSE  NULL,
int570                  int               SPARSE  NULL,
int571                  int               SPARSE  NULL,
<table>
<thead>
<tr>
<th>Index</th>
<th>Type</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>int572</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int573</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int574</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int575</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int576</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int577</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int578</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int579</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int580</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int581</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int582</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int583</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int584</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int585</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int586</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int587</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int588</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int589</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int590</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int591</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int592</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int593</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int594</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int595</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int596</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int597</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int598</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int599</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int600</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int601</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int602</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int603</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int604</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int605</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int606</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int607</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int608</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int609</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int610</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int611</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int612</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int613</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int614</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int615</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int616</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int617</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int618</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int619</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int620</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int621</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int622</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int623</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int624</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int625</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int626</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int627</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int628</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int629</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int630</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int631</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int632</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int633</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int634</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int635</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int636</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int637</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int638</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int639</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int640</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int641</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int642</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int643</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int644</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int645</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int646</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int647</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int648</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int649</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int650</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int651</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int652</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int653</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int654</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int655</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int656</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int657</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int658</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int659</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int660</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int661</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int662</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int663</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int664</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int665</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int666</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int667</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int668</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int669</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int670</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int671</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int672</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int673</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int674</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int675</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int676</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int677</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int678</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int679</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int680</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int681</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int682</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int683</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int684</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int685</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int686</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int687</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int688</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int689</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int690</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int691</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int692</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int693</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int694</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int695</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int696</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int697</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int698</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int699</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int700</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int701</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int702</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int703</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int704</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int705</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int706</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int707</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int708</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>int709</td>
<td>int</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
</tbody>
</table>
int710                  int               SPARSE  NULL,
int711                  int               SPARSE  NULL,
int712                  int               SPARSE  NULL,
int713                  int               SPARSE  NULL,
int714                  int               SPARSE  NULL,
int715                  int               SPARSE  NULL,
int716                  int               SPARSE  NULL,
int717                  int               SPARSE  NULL,
int718                  int               SPARSE  NULL,
int719                  int               SPARSE  NULL,
int720                  int               SPARSE  NULL,
int721                  int               SPARSE  NULL,
int722                  int               SPARSE  NULL,
int723                  int               SPARSE  NULL,
int724                  int               SPARSE  NULL,
int725                  int               SPARSE  NULL,
int726                  int               SPARSE  NULL,
int727                  int               SPARSE  NULL,
int728                  int               SPARSE  NULL,
int729                  int               SPARSE  NULL,
int730                  int               SPARSE  NULL,
int731                  int               SPARSE  NULL,
int732                  int               SPARSE  NULL,
int733                  int               SPARSE  NULL,
int734                  int               SPARSE  NULL,
int735                  int               SPARSE  NULL,
int736                  int               SPARSE  NULL,
int737                  int               SPARSE  NULL,
int738                  int               SPARSE  NULL,
int739                  int               SPARSE  NULL,
int740                  int               SPARSE  NULL,
int741                  int               SPARSE  NULL,
int742                  int               SPARSE  NULL,
int743                  int               SPARSE  NULL,
int744                  int               SPARSE  NULL,
int745                  int               SPARSE  NULL,
int746                  int               SPARSE  NULL,
int747                  int               SPARSE  NULL,
int748                  int               SPARSE  NULL,
int749                  int               SPARSE  NULL,
int750                  int               SPARSE  NULL,
text1                  nvarchar(max)     SPARSE  NULL,
text2                  nvarchar(max)     SPARSE  NULL,
text3                  nvarchar(max)     SPARSE  NULL,
text4                  nvarchar(max)     SPARSE  NULL,
text5                  nvarchar(max)     SPARSE  NULL,
text6                  nvarchar(max)     SPARSE  NULL,
text7                  nvarchar(max)     SPARSE  NULL,
text8                  nvarchar(max)     SPARSE  NULL,
text9                  nvarchar(max)     SPARSE  NULL,
text10                 nvarchar(max)     SPARSE  NULL,
text11                 nvarchar(max)     SPARSE  NULL,
text12                 nvarchar(max)     SPARSE  NULL,
text13                 nvarchar(max)     SPARSE  NULL,
text14                 nvarchar(max)     SPARSE  NULL,
text15                 nvarchar(max)     SPARSE  NULL,
text16                 nvarchar(max)     SPARSE  NULL,
text17                 nvarchar(max)     SPARSE  NULL,
text18                 nvarchar(max)     SPARSE  NULL,
text19                 nvarchar(max)     SPARSE  NULL,
text20                 nvarchar(max)     SPARSE  NULL,
text21                 nvarchar(max)     SPARSE  NULL,
text22                 nvarchar(max)     SPARSE  NULL,
text23                 nvarchar(max)     SPARSE  NULL,
text24                 nvarchar(max)     SPARSE  NULL,
text25                 nvarchar(max)     SPARSE  NULL,
text26                 nvarchar(max)     SPARSE  NULL,
text27                 nvarchar(max)     SPARSE  NULL,
text28                 nvarchar(max)     SPARSE  NULL,
<table>
<thead>
<tr>
<th>ntext98</th>
<th>nvarchar(max)</th>
<th>SPARSE NULL,</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntext99</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext100</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext101</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext102</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext103</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext104</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext105</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext106</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext107</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext108</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext109</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext110</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext111</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext112</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext113</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext114</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext115</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext116</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext117</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext118</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext119</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext120</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext121</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext122</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext123</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext124</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext125</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext126</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext127</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext128</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext129</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext130</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext131</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext132</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext133</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext134</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext135</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext136</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext137</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext138</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext139</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext140</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext141</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext142</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext143</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext144</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext145</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext146</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext147</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext148</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext149</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext150</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext151</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext152</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext153</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext154</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext155</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext156</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext157</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext158</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext159</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext160</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext161</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext162</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext163</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext164</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext165</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext166</td>
<td>nvarchar(max)</td>
<td>SPARSE NULL,</td>
</tr>
<tr>
<td>ntext236</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext237</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext238</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext239</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext240</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext241</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext242</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext243</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext244</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext245</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext246</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext247</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext248</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext249</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext250</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext251</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext252</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext253</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext254</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext255</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext256</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext257</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext258</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext259</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext260</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext261</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>ntext262</td>
<td>nvarchar(max)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar1</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar2</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar3</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar4</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar5</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar6</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar7</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar8</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar9</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar10</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar11</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar12</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar13</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar14</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar15</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar16</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar17</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar18</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar19</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar20</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar21</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar22</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar23</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar24</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar25</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar26</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar27</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar28</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar29</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar30</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar31</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar32</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar33</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar34</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar35</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar36</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar37</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar38</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar39</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar40</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar41</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
<tr>
<td>nvarchar42</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
</tr>
</tbody>
</table>
nvarchar43  nvarchar (255)  SPARSE  NULL,
nvarchar44  nvarchar (255)  SPARSE  NULL,
nvarchar45  nvarchar (255)  SPARSE  NULL,
nvarchar46  nvarchar (255)  SPARSE  NULL,
nvarchar47  nvarchar (255)  SPARSE  NULL,
nvarchar48  nvarchar (255)  SPARSE  NULL,
nvarchar49  nvarchar (255)  SPARSE  NULL,
nvarchar50  nvarchar (255)  SPARSE  NULL,
nvarchar51  nvarchar (255)  SPARSE  NULL,
nvarchar52  nvarchar (255)  SPARSE  NULL,
nvarchar53  nvarchar (255)  SPARSE  NULL,
nvarchar54  nvarchar (255)  SPARSE  NULL,
nvarchar55  nvarchar (255)  SPARSE  NULL,
nvarchar56  nvarchar (255)  SPARSE  NULL,
nvarchar57  nvarchar (255)  SPARSE  NULL,
nvarchar58  nvarchar (255)  SPARSE  NULL,
nvarchar59  nvarchar (255)  SPARSE  NULL,
nvarchar60  nvarchar (255)  SPARSE  NULL,
nvarchar61  nvarchar (255)  SPARSE  NULL,
nvarchar62  nvarchar (255)  SPARSE  NULL,
nvarchar63  nvarchar (255)  SPARSE  NULL,
nvarchar64  nvarchar (255)  SPARSE  NULL,
nvarchar65  nvarchar (255)  SPARSE  NULL,
nvarchar66  nvarchar (255)  SPARSE  NULL,
nvarchar67  nvarchar (255)  SPARSE  NULL,
nvarchar68  nvarchar (255)  SPARSE  NULL,
nvarchar69  nvarchar (255)  SPARSE  NULL,
nvarchar70  nvarchar (255)  SPARSE  NULL,
nvarchar71  nvarchar (255)  SPARSE  NULL,
nvarchar72  nvarchar (255)  SPARSE  NULL,
nvarchar73  nvarchar (255)  SPARSE  NULL,
nvarchar74  nvarchar (255)  SPARSE  NULL,
nvarchar75  nvarchar (255)  SPARSE  NULL,
nvarchar76  nvarchar (255)  SPARSE  NULL,
nvarchar77  nvarchar (255)  SPARSE  NULL,
nvarchar78  nvarchar (255)  SPARSE  NULL,
nvarchar79  nvarchar (255)  SPARSE  NULL,
nvarchar80  nvarchar (255)  SPARSE  NULL,
nvarchar81  nvarchar (255)  SPARSE  NULL,
nvarchar82  nvarchar (255)  SPARSE  NULL,
nvarchar83  nvarchar (255)  SPARSE  NULL,
nvarchar84  nvarchar (255)  SPARSE  NULL,
nvarchar85  nvarchar (255)  SPARSE  NULL,
nvarchar86  nvarchar (255)  SPARSE  NULL,
nvarchar87  nvarchar (255)  SPARSE  NULL,
nvarchar88  nvarchar (255)  SPARSE  NULL,
nvarchar89  nvarchar (255)  SPARSE  NULL,
nvarchar90  nvarchar (255)  SPARSE  NULL,
nvarchar91  nvarchar (255)  SPARSE  NULL,
nvarchar92  nvarchar (255)  SPARSE  NULL,
nvarchar93  nvarchar (255)  SPARSE  NULL,
nvarchar94  nvarchar (255)  SPARSE  NULL,
nvarchar95  nvarchar (255)  SPARSE  NULL,
nvarchar96  nvarchar (255)  SPARSE  NULL,
nvarchar97  nvarchar (255)  SPARSE  NULL,
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>SPARSE</th>
<th>NULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>nvarchar112</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar113</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar114</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar115</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar116</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar117</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar118</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar119</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar120</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar121</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar122</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar123</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar124</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar125</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar126</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar127</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar128</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar129</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar130</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar131</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar132</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar133</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar134</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar135</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar136</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar137</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar138</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar139</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar140</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar141</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar142</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar143</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar144</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar145</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar146</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar147</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar148</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar149</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar150</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar151</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar152</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar153</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar154</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar155</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar156</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar157</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar158</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar159</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar160</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar161</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar162</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar163</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar164</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar165</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar166</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar167</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar168</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar169</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar170</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar171</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar172</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar173</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar174</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar175</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar176</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar177</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar178</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar179</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar180</td>
<td>nvarchar(255)</td>
<td>SPARSE</td>
<td>NULL</td>
</tr>
<tr>
<td>nvarchar181</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar182</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar183</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar184</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar185</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar186</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar187</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar188</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar189</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar190</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar191</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar192</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar193</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar194</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar195</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar196</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar197</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar198</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar199</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar200</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar201</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar202</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar203</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar204</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar205</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar206</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar207</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar208</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar209</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar210</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar211</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar212</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar213</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar214</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar215</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar216</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar217</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar218</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar219</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar220</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar221</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar222</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar223</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar224</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar225</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar226</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar227</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar228</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar229</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar230</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar231</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar232</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar233</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar234</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar235</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar236</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar237</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar238</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar239</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar240</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar241</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar242</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar243</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar244</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar245</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar246</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar247</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar248</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
<tr>
<td>nvarchar249</td>
<td>nvarchar(255)</td>
<td>SPARSE NULL,</td>
<td></td>
</tr>
</tbody>
</table>
nvarchar250  nvarchar(255)     SPARSE  NULL,
nvarchar251  nvarchar(255)     SPARSE  NULL,
nvarchar252  nvarchar(255)     SPARSE  NULL,
nvarchar253  nvarchar(255)     SPARSE  NULL,
nvarchar254  nvarchar(255)     SPARSE  NULL,
nvarchar255  nvarchar(255)     SPARSE  NULL,
nvarchar256  nvarchar(255)     SPARSE  NULL,
nvarchar257  nvarchar(255)     SPARSE  NULL,
nvarchar258  nvarchar(255)     SPARSE  NULL,
nvarchar259  nvarchar(255)     SPARSE  NULL,
nvarchar260  nvarchar(255)     SPARSE  NULL,
nvarchar261  nvarchar(255)     SPARSE  NULL,
nvarchar262  nvarchar(255)     SPARSE  NULL,
sql_variant1  sql_variant     SPARSE  NULL,
sql_variant2  sql_variant     SPARSE  NULL,
sql_variant3  sql_variant     SPARSE  NULL,
sql_variant4  sql_variant     SPARSE  NULL,
sql_variant5  sql_variant     SPARSE  NULL,
sql_variant6  sql_variant     SPARSE  NULL,
sql_variant7  sql_variant     SPARSE  NULL,
sql_variant8  sql_variant     SPARSE  NULL,
sql_variant9  sql_variant     SPARSE  NULL,
sql_variant10  sql_variant     SPARSE  NULL,
sql_variant11  sql_variant     SPARSE  NULL,
sql_variant12  sql_variant     SPARSE  NULL,
sql_variant13  sql_variant     SPARSE  NULL,
sql_variant14  sql_variant     SPARSE  NULL,
sql_variant15  sql_variant     SPARSE  NULL,
sql_variant16  sql_variant     SPARSE  NULL,
sql_variant17  sql_variant     SPARSE  NULL,
sql_variant18  sql_variant     SPARSE  NULL,
sql_variant19  sql_variant     SPARSE  NULL,
sql_variant20  sql_variant     SPARSE  NULL,
sql_variant21  sql_variant     SPARSE  NULL,
sql_variant22  sql_variant     SPARSE  NULL,
sql_variant23  sql_variant     SPARSE  NULL,
sql_variant24  sql_variant     SPARSE  NULL,
sql_variant25  sql_variant     SPARSE  NULL,
sql_variant26  sql_variant     SPARSE  NULL,
sql_variant27  sql_variant     SPARSE  NULL,
sql_variant28  sql_variant     SPARSE  NULL,
sql_variant29  sql_variant     SPARSE  NULL,
sql_variant30  sql_variant     SPARSE  NULL,
| sql_variant57 | sql_variant | SPARSE | NULL, |
| sql_variant58 | sql_variant | SPARSE | NULL, |
| sql_variant59 | sql_variant | SPARSE | NULL, |
| sql_variant60 | sql_variant | SPARSE | NULL, |
| sql_variant61 | sql_variant | SPARSE | NULL, |
| sql_variant62 | sql_variant | SPARSE | NULL, |
| sql_variant63 | sql_variant | SPARSE | NULL, |
| sql_variant64 | sql_variant | SPARSE | NULL, |
| sql_variant65 | sql_variant | SPARSE | NULL, |
| sql_variant66 | sql_variant | SPARSE | NULL, |
| sql_variant67 | sql_variant | SPARSE | NULL, |
| sql_variant68 | sql_variant | SPARSE | NULL, |
| sql_variant69 | sql_variant | SPARSE | NULL, |
| sql_variant70 | sql_variant | SPARSE | NULL, |
| sql_variant71 | sql_variant | SPARSE | NULL, |
| sql_variant72 | sql_variant | SPARSE | NULL, |
| sql_variant73 | sql_variant | SPARSE | NULL, |
| sql_variant74 | sql_variant | SPARSE | NULL, |
| sql_variant75 | sql_variant | SPARSE | NULL, |
| sql_variant76 | sql_variant | SPARSE | NULL, |
| sql_variant77 | sql_variant | SPARSE | NULL, |
| sql_variant78 | sql_variant | SPARSE | NULL, |
| sql_variant79 | sql_variant | SPARSE | NULL, |
| sql_variant80 | sql_variant | SPARSE | NULL, |
| sql_variant81 | sql_variant | SPARSE | NULL, |
| sql_variant82 | sql_variant | SPARSE | NULL, |
| sql_variant83 | sql_variant | SPARSE | NULL, |
| sql_variant84 | sql_variant | SPARSE | NULL, |
| sql_variant85 | sql_variant | SPARSE | NULL, |
| sql_variant86 | sql_variant | SPARSE | NULL, |
| sql_variant87 | sql_variant | SPARSE | NULL, |
| sql_variant88 | sql_variant | SPARSE | NULL, |
| sql_variant89 | sql_variant | SPARSE | NULL, |
| sql_variant90 | sql_variant | SPARSE | NULL, |
| sql_variant91 | sql_variant | SPARSE | NULL, |
| sql_variant92 | sql_variant | SPARSE | NULL, |
| sql_variant93 | sql_variant | SPARSE | NULL, |
| sql_variant94 | sql_variant | SPARSE | NULL, |
| sql_variant95 | sql_variant | SPARSE | NULL, |
| sql_variant96 | sql_variant | SPARSE | NULL, |
| sql_variant97 | sql_variant | SPARSE | NULL, |
| sql_variant98 | sql_variant | SPARSE | NULL, |
| sql_variant99 | sql_variant | SPARSE | NULL, |
| sql_variant100 | sql_variant | SPARSE | NULL, |
| sql_variant101 | sql_variant | SPARSE | NULL, |
| sql_variant102 | sql_variant | SPARSE | NULL, |
| sql_variant103 | sql_variant | SPARSE | NULL, |
| sql_variant104 | sql_variant | SPARSE | NULL, |
| sql_variant105 | sql_variant | SPARSE | NULL, |
| sql_variant106 | sql_variant | SPARSE | NULL, |
| sql_variant107 | sql_variant | SPARSE | NULL, |
| sql_variant108 | sql_variant | SPARSE | NULL, |
| sql_variant109 | sql_variant | SPARSE | NULL, |
| sql_variant110 | sql_variant | SPARSE | NULL, |
| sql_variant111 | sql_variant | SPARSE | NULL, |
| sql_variant112 | sql_variant | SPARSE | NULL, |
| sql_variant113 | sql_variant | SPARSE | NULL, |
| sql_variant114 | sql_variant | SPARSE | NULL, |
| sql_variant115 | sql_variant | SPARSE | NULL, |
| sql_variant116 | sql_variant | SPARSE | NULL, |
| sql_variant117 | sql_variant | SPARSE | NULL, |
| sql_variant118 | sql_variant | SPARSE | NULL, |
| sql_variant119 | sql_variant | SPARSE | NULL, |
| sql_variant120 | sql_variant | SPARSE | NULL, |
| sql_variant121 | sql_variant | SPARSE | NULL, |
| sql_variant122 | sql_variant | SPARSE | NULL, |
| sql_variant123 | sql_variant | SPARSE | NULL, |
| sql_variant124 | sql_variant | SPARSE | NULL, |
| sql_variant125 | sql_variant | SPARSE | NULL,
<table>
<thead>
<tr>
<th>sql_variant126</th>
<th>sql_variant</th>
<th>SPARSE</th>
<th>NULL,</th>
</tr>
</thead>
<tbody>
<tr>
<td>sql_variant127</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant128</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant129</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant130</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant131</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant132</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant133</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant134</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant135</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant136</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant137</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant138</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant139</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant140</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant141</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant142</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant143</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant144</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant145</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant146</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant147</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant148</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant149</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant150</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant151</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant152</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant153</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant154</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant155</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant156</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant157</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant158</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant159</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant160</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant161</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant162</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant163</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant164</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant165</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant166</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant167</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant168</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant169</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant170</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant171</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant172</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant173</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant174</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant175</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant176</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant177</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant178</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant179</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant180</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant181</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant182</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant183</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant184</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant185</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant186</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant187</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant188</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant189</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant190</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant191</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant192</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant193</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant194</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant195</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>sql_variant196</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant197</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant198</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant199</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant200</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant201</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant202</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant203</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant204</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant205</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant206</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant207</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant208</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant209</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant210</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant211</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant212</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant213</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant214</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant215</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant216</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant217</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant218</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant219</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant220</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant221</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant222</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant223</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant224</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant225</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant226</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant227</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant228</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant229</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant230</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant231</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant232</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant233</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant234</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant235</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant236</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant237</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant238</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant239</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant240</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant241</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant242</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant243</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant244</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant245</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant246</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant247</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant248</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant249</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant250</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant251</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant252</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant253</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant254</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant255</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant256</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant257</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant258</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant259</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant260</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant261</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>sql_variant262</td>
<td>sql_variant</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
<tr>
<td>uniqueidentifier1</td>
<td>uniqueidentifier</td>
<td>SPARSE</td>
<td>NULL,</td>
</tr>
</tbody>
</table>
uniqueidentifier2   uniqueidentifier   SPARSE   NULL,
uniqueidentifier3   uniqueidentifier   SPARSE   NULL,
uniqueidentifier4   uniqueidentifier   SPARSE   NULL,
uniqueidentifier5   uniqueidentifier   SPARSE   NULL,
uniqueidentifier6   uniqueidentifier   SPARSE   NULL,
uniqueidentifier7   uniqueidentifier   SPARSE   NULL,
uniqueidentifier8   uniqueidentifier   SPARSE   NULL,
uniqueidentifier9   uniqueidentifier   SPARSE   NULL,
uniqueidentifier10  uniqueidentifier   SPARSE   NULL,
uniqueidentifier11  uniqueidentifier   SPARSE   NULL,
uniqueidentifier12  uniqueidentifier   SPARSE   NULL,
uniqueidentifier13  uniqueidentifier   SPARSE   NULL,
uniqueidentifier14  uniqueidentifier   SPARSE   NULL,
uniqueidentifier15  uniqueidentifier   SPARSE   NULL,
uniqueidentifier16  uniqueidentifier   SPARSE   NULL,
uniqueidentifier17  uniqueidentifier   SPARSE   NULL,
uniqueidentifier18  uniqueidentifier   SPARSE   NULL,
uniqueidentifier19  uniqueidentifier   SPARSE   NULL,
uniqueidentifier20  uniqueidentifier   SPARSE   NULL,
uniqueidentifier21  uniqueidentifier   SPARSE   NULL,
uniqueidentifier22  uniqueidentifier   SPARSE   NULL,
uniqueidentifier23  uniqueidentifier   SPARSE   NULL,
uniqueidentifier24  uniqueidentifier   SPARSE   NULL,
uniqueidentifier25  uniqueidentifier   SPARSE   NULL,
uniqueidentifier26  uniqueidentifier   SPARSE   NULL,
uniqueidentifier27  uniqueidentifier   SPARSE   NULL,
uniqueidentifier28  uniqueidentifier   SPARSE   NULL,
uniqueidentifier29  uniqueidentifier   SPARSE   NULL,
uniqueidentifier30  uniqueidentifier   SPARSE   NULL,
uniqueidentifier31  uniqueidentifier   SPARSE   NULL,
uniqueidentifier32  uniqueidentifier   SPARSE   NULL,
uniqueidentifier33  uniqueidentifier   SPARSE   NULL,
uniqueidentifier34  uniqueidentifier   SPARSE   NULL,
uniqueidentifier35  uniqueidentifier   SPARSE   NULL,
uniqueidentifier36  uniqueidentifier   SPARSE   NULL,
uniqueidentifier37  uniqueidentifier   SPARSE   NULL,
uniqueidentifier38  uniqueidentifier   SPARSE   NULL,
uniqueidentifier39  uniqueidentifier   SPARSE   NULL,
uniqueidentifier40  uniqueidentifier   SPARSE   NULL,
uniqueidentifier41  uniqueidentifier   SPARSE   NULL,
uniqueidentifier42  uniqueidentifier   SPARSE   NULL,
uniqueidentifier43  uniqueidentifier   SPARSE   NULL,
uniqueidentifier44  uniqueidentifier   SPARSE   NULL,
uniqueidentifier140  uniqueidentifier SPARSE NULL,
uniqueidentifier141  uniqueidentifier SPARSE NULL,
uniqueidentifier142  uniqueidentifier SPARSE NULL,
uniqueidentifier143  uniqueidentifier SPARSE NULL,
uniqueidentifier144  uniqueidentifier SPARSE NULL,
uniqueidentifier145  uniqueidentifier SPARSE NULL,
uniqueidentifier146  uniqueidentifier SPARSE NULL,
uniqueidentifier147  uniqueidentifier SPARSE NULL,
uniqueidentifier148  uniqueidentifier SPARSE NULL,
uniqueidentifier149  uniqueidentifier SPARSE NULL,
uniqueidentifier150  uniqueidentifier SPARSE NULL,
uniqueidentifier151  uniqueidentifier SPARSE NULL,
uniqueidentifier152  uniqueidentifier SPARSE NULL,
uniqueidentifier153  uniqueidentifier SPARSE NULL,
uniqueidentifier154  uniqueidentifier SPARSE NULL,
uniqueidentifier155  uniqueidentifier SPARSE NULL,
uniqueidentifier156  uniqueidentifier SPARSE NULL,
uniqueidentifier157  uniqueidentifier SPARSE NULL,
uniqueidentifier158  uniqueidentifier SPARSE NULL,
uniqueidentifier159  uniqueidentifier SPARSE NULL,
uniqueidentifier160  uniqueidentifier SPARSE NULL,
uniqueidentifier161  uniqueidentifier SPARSE NULL,
uniqueidentifier162  uniqueidentifier SPARSE NULL,
uniqueidentifier163  uniqueidentifier SPARSE NULL,
uniqueidentifier164  uniqueidentifier SPARSE NULL,
uniqueidentifier165  uniqueidentifier SPARSE NULL,
uniqueidentifier166  uniqueidentifier SPARSE NULL,
uniqueidentifier167  uniqueidentifier SPARSE NULL,
uniqueidentifier168  uniqueidentifier SPARSE NULL,
uniqueidentifier169  uniqueidentifier SPARSE NULL,
uniqueidentifier170  uniqueidentifier SPARSE NULL,
uniqueidentifier171  uniqueidentifier SPARSE NULL,
uniqueidentifier172  uniqueidentifier SPARSE NULL,
uniqueidentifier173  uniqueidentifier SPARSE NULL,
uniqueidentifier174  uniqueidentifier SPARSE NULL,
uniqueidentifier175  uniqueidentifier SPARSE NULL,
uniqueidentifier176  uniqueidentifier SPARSE NULL,
uniqueidentifier177  uniqueidentifier SPARSE NULL,
uniqueidentifier178  uniqueidentifier SPARSE NULL,
uniqueidentifier179  uniqueidentifier SPARSE NULL,
uniqueidentifier180  uniqueidentifier SPARSE NULL,
uniqueidentifier181  uniqueidentifier SPARSE NULL,
uniqueidentifier182  uniqueidentifier SPARSE NULL,
uniqueidentifier183  uniqueidentifier SPARSE NULL,
uniqueidentifier184  uniqueidentifier SPARSE NULL,
uniqueidentifier185  uniqueidentifier SPARSE NULL,
uniqueidentifier186  uniqueidentifier SPARSE NULL,
uniqueidentifier187  uniqueidentifier SPARSE NULL,
uniqueidentifier188  uniqueidentifier SPARSE NULL,
uniqueidentifier189  uniqueidentifier SPARSE NULL,
uniqueidentifier190  uniqueidentifier SPARSE NULL,
uniqueidentifier191  uniqueidentifier SPARSE NULL,
uniqueidentifier192  uniqueidentifier SPARSE NULL,
uniqueidentifier193  uniqueidentifier SPARSE NULL,
uniqueidentifier194  uniqueidentifier SPARSE NULL,
uniqueidentifier195  uniqueidentifier SPARSE NULL,
uniqueidentifier196  uniqueidentifier SPARSE NULL,
uniqueidentifier197  uniqueidentifier SPARSE NULL,
uniqueidentifier198  uniqueidentifier SPARSE NULL,
uniqueidentifier199  uniqueidentifier SPARSE NULL,
uniqueidentifier200  uniqueidentifier SPARSE NULL,
uniqueidentifier201  uniqueidentifier SPARSE NULL,
uniqueidentifier202  uniqueidentifier SPARSE NULL,
uniqueidentifier203  uniqueidentifier SPARSE NULL,
uniqueidentifier204  uniqueidentifier SPARSE NULL,
uniqueidentifier205  uniqueidentifier SPARSE NULL,
uniqueidentifier206  uniqueidentifier SPARSE NULL,
uniqueidentifier207  uniqueidentifier SPARSE NULL,
uniqueidentifier208  uniqueidentifier SPARSE NULL,
| uniqueidentifier278 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier279 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier280 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier281 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier282 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier283 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier284 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier285 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier286 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier287 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier288 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier289 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier290 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier291 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier292 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier293 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier294 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier295 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier296 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier297 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier298 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier299 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier300 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier301 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier302 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier303 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier304 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier305 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier306 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier307 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier308 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier309 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier310 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier311 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier312 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier313 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier314 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier315 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier316 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier317 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier318 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier319 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier320 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier321 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier322 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier323 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier324 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier325 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier326 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier327 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier328 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier329 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier330 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier331 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier332 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier333 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier334 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier335 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier336 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier337 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier338 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier339 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier340 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier341 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier342 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier343 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier344 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier345 | uniqueidentifier | SPARSE | NULL, |
| uniqueidentifier346 | uniqueidentifier | SPARSE | NULL, |
tp_Id: The identifier for the list item, uniquely identifying it within the AllUserData table.

tp_ListId: The list identifier of the list (1) or document library containing the list item.

tp_SiteId: The site collection identifier of the site collection containing the list item.

tp_RowOrdinal: The zero-based ordinal index of this row in the set of rows representing the list item. Additional rows are used to represent list items that have more application-defined columns of one or more data types than can fit in a single row in the AllUserData table.

tp_Version: A counter used for internal conflict detection that is incremented any time a change is made to the list item. Because of the mapping of application properties to the generic columns schema in this table, changes to application schema as well as property values can affect a version increment.

tp_Author: The user identifier for the user who created the list item.

tp_Editor: The user identifier for the user who last edited the list item.

tp_Modified: A date and time value in UTC format specifying when this list item was last modified.

tp_Created: A date and time value in UTC format specifying when this list item was created.

tp_Ordering: A concatenation of time stamp values in yyyyMMddHHmmss format, specifying the threading structure of the list items in a legacy discussion board list (1) (a list (1) with a List Base Type (section 2.2.1.2.11) of three). For list items in all other types of lists (1), this parameter MUST be NULL.

tp_ThreadIndex: A binary structure specifying the list item’s position within a legacy discussion board list (1) (a list (1) with a List Base Type of three). For list items in all other types of lists (1), this parameter MUST be NULL.

tp_HasAttachment: A bit set to one if the list item has an attachment associated with it; otherwise, it is set to zero.

tp_ModerationStatus: A Moderation Status value (section 2.2.1.2.13) indicating the current moderation approval status of the list item.

tp_IsCurrent: A bit set to one if this is a current version of this list item; otherwise, it is set to zero.

tp_ItemOrder: A value used to calculate the relative order in which to view the list item when displayed with other list items from the same list (1).

tp_InstanceID: If this list item is associated with a particular instance of a recurring meeting, this is the integer identifier of that instance. For all other list items, this MUST be NULL.

tp_GUID: A value uniquely identifying this list item.

tp_CopySource: The URL used as a source for the list item. If this list item was not copied from a source list item, this value MUST be NULL.

tp_HasCopyDestinations: This bit is set to one if destination locations for the list item to be copied to have been set. If the list item does not have a destination location set, this value MUST be zero.
tp_AuditFlags: An Audit Flags value (section 2.2.1) determining the operations to be tracked on this list item.

tp_InheritAuditFlags: An Audit Flags value for the operations to be tracked on this list item, as determined from parent container Audit Flags settings.

tp_Size: The sum of the size, in bytes, of the content of application-schema columns in the list item. This does not include the size of the document stream for list items that have an associated document stream.

tp_WorkflowVersion: If the list item is part of a workflow, this stores an integer denoting the state of this list item within that workflow. Otherwise, this value MUST be NULL.

tp_WorkflowInstanceID: A Workflow Identifier value (section 2.2.1.1.16) for the currently active workflow instance on this list item. If the list item is not part of a workflow, this value MUST be NULL.

tp_ParentId: The document identifier of the item's parent container. For example, the document identifier of the folder or document library containing this subfolder or item. This value MUST NOT be NULL, because every item but one has a parent container. A special empty document identifier, '00000000-0000-0000-0000-000000000000', marks the parent container of the topmost item (the root site (2)) in the site collection.

tp_DocId: The document identifier of the list item.

tp_DeleteTransactionId: An identifier for use with an implementation-specific deleted items Recycle Bin. This MUST equal 0x if the list item is nondeleted.

tp_ContentTypeId: The binary identifier of the content type associated with the list item.

ContentTypeId is defined in section 2.2.1.1.

tp_Level: A publishing level value specifying the publishing status of this version of the list item.

tp_IsCurrentVersion: A bit indicating whether this row corresponds to a current version or a historical version of the list item. This value MUST be one if this row contains a current version. Otherwise, it MUST be zero.

tp_UIVersion: The UI version number associated with this list item. The default value of tp_UIVersion is 512, which corresponds to a displayed version of 1.0.

tp_CalculatedVersion: This contains the UI version number if this is a historical version of the list item. This MUST be zero if the list item is a current version.

tp_UIVersionString: A calculated column containing the value of the tp_UIVersion column as a displayed version string.

tp_DraftOwnerId: The identifier of the user who created this list item as a draft. This value is non-NULL only if the list item exists and is a draft version.

tp_CheckoutUserId: The identifier of the user who checked out this list item. This value is non-NULL only if the list item exists and is checked out.

tp_AppAuthor: The app principal identifier of the app principal who created the list item.

tp_AppEditor: The app principal identifier of the app principal who last edited the list item.

The next nine columns are duplicated a number of times within the table definition. This is indicated using a suffix, #, which is replaced with a numeral in the column names, described as follows. The number of times each column is duplicated varies and is indicated for each column. Each group of columns is dedicated to hold application-defined fields of a different data type, described as follows.
bit#: Columns for application-defined fields that hold values of type bit. The 1000 columns are named bit1 to bit1000. If the column does not contain data, this value MUST be NULL.

datetime#: Columns for application-defined fields that hold values of type datetime. The 550 columns are named datetime1 to datetime550. If the column does not contain data, this value MUST be NULL.

float#: Columns for application-defined fields that hold values of type float. The 550 columns are named float1 to float550. If the column does not contain data, this value MUST be NULL.

int#: Columns for application-defined fields that hold values of type int. The 750 columns are named int1 to int750. If the column does not contain data, this value MUST be NULL.

nvarchar#: Columns for application-defined fields that hold values of type nvarchar. The 262 columns are named nvarchar1 to nvarchar262. If the column does not contain data, this value MUST be NULL.

sql_variant#: Columns for application-defined fields that hold values of type sql_variant. The 262 columns are named sql_variant1 to sql_variant262. If the column does not contain data, this value MUST be NULL.

uniqueidentifier#: Columns for application-defined fields that hold values of type uniqueidentifier. The 350 columns are named uniqueidentifier1 to uniqueidentifier350. If the column does not contain data, this value MUST be NULL.

geography#: Columns for application-defined fields that hold values of type varbinary(8000). The 2 columns are named geography1 and geography2. If the column does not contain data, this value MUST be NULL.

tp_ColumnSet: An XML representation of the non-NULL values from the columns bit#, datetime#, float#, int#, ntext#, nvarchar#, sql_variant#, uniqueidentifier#, and geography#.

2.2.6.3 Docs View

The Docs View reflects all rows in the table AllDocs (section 2.2.6.1) that are not deleted (that is, all table rows where DeleteTransactionId is equal to 0x). Docs View records include items that have a URL, such as files, folders, lists, document libraries, and sites, referred to collectively as documents.

See AllDocs for the T-SQL syntax description of the structure of this view.

2.2.6.4 GroupMembership Table

The GroupMembership Table stores a list of the principals that are members of each site group in a site collection. The GroupMembership Table is defined using T-SQL syntax, as follows.

```
TABLE GroupMembership(
    SiteId uniqueidentifier NOT NULL,
    GroupId int NOT NULL,
    MemberId int NOT NULL
);
```

SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the principals and site groups.
**GroupId**: The group identifier of the site group that the principal (1) specified by **MemberId** is a member of.

**MemberId**: The User Identifier (section 2.2.1.13) of a principal that is a member of the site group specified by **GroupId**.

### 2.2.6.5 Sec_SiteGroupsView

The **Sec_SiteGroupsView** is a view into site group information with one site group per row. Site groups available through this view are owned by a user or domain group, or by a site group.

If a user or domain group owns the site group, the **OwnerIsUser** bit is set to one, and the **GroupId**, **GroupName**, **GroupDescription**, **GroupSiteID**, **GroupOwner** and **GroupOwnerIsUser** fields are set to NULL. If the site group is owned by a site group, the **OwnerIsUser** bit is set to zero, and the **UserID**, **UserSID**, **UserName**, **UserLogin**, **UserEmail**, **UserNotes**, **UserSiteAdmin** and **UserDomainGroup** are set to NULL.

The **Sec_SiteGroupsView** is defined using T-SQL syntax, as follows.

```sql
VIEW Sec_SiteGroupsView(
    ID int,
    Title nvarchar(255),
    Description nvarchar(512),
    SiteWebId uniqueidentifier,
    Owner int,
    OwnerIsUser bit,
    OwnerGlobal bit,
    Type tinyint,
    Global bit,
    SiteID uniqueidentifier,
    UserID int,
    UserID nvarchar(255),
    UserName nvarchar(255),
    UserLogin nvarchar(255),
    UserEmail nvarchar(255),
    UserNotes nvarchar(1023),
    UserSiteAdmin bit,
    UserDomainGroup bit,
    UserFlags int,
    GroupID int,
    GroupName nvarchar(255),
    GroupDescription nvarchar(512),
    GroupSiteID uniqueidentifier,
    GroupOwner int,
    GroupOwnerIsUser bit,
    DLAlias nvarchar(128),
    DLErrorMessage nvarchar(512),
    DLFlags int,
    DLJobId int,
    DLArchives varbinary(4000),
    RequestEmail nvarchar(255),
    Flags int,
);
```

**ID**: The identifier for the site group. This parameter MUST NOT be NULL.

**Title**: The user-friendly display name for the site group. This parameter MUST NOT be NULL.

**Description**: The site group description.

**SiteWebId**: The **Site Collection Identifier** (section 2.2.1.9) of the **site collection** that contains the site group information. This value is the same as returned in **SiteId**. For non-user-owned groups, this column name will be **WebId**. This parameter MUST NOT be NULL.

---

[MS-WSSFO3] - v20170620
Windows SharePoint Services (WSS): File Operations Database Communications Version 3 Protocol
Copyright © 2017 Microsoft Corporation
Release: June 20, 2017
**Owner**: The **User Identifier** (section 2.2.1.13) or **Site Group Identifier** (section 2.2.1.10) of the site group owner. This parameter MUST NOT be NULL.

**OwnerIsUser**: A bit flag specifying whether the site group owner is a user or a site group. When the value in the **Owner field** is a **User Identifier** for a user or a domain group, the **OwnerIsUser** flag MUST be set to one. When the value in the **Owner field** is a **Site Group Identifier**, the **OwnerIsUser** flag MUST be set to zero. This parameter MUST NOT be NULL.

**OwnerGlobal**: A bit flag specifying whether the view contains ownership information for a user or domain group, or for a site group.

When the owner is a site group, the **OwnerGlobal** flag MUST be set to one and the **GroupID**, **GroupName**, **GroupDescription**, **GroupSiteID**, **GroupOwner** and **GroupOwnerIsUser** fields MUST be populated with information from the site group owning this site group, while the **UserID**, **UserSID**, **UserName**, **UserLogin**, **UserEmail**, **UserNotes**, **UserSiteAdmin** and **UserDomainGroup** fields MUST be NULL.

When the owner is a user or domain group, the **OwnerGlobal** flag MUST be set to zero and the **UserID**, **UserSID**, **UserName**, **UserLogin**, **UserEmail**, **UserNotes**, **UserSiteAdmin** and **UserDomainGroup** fields MUST be populated with information from the user or domain group owning this site group, while the **GroupID**, **GroupName**, **GroupDescription**, **GroupSiteID**, **GroupOwner** and **GroupOwnerIsUser** fields MUST be NULL.

**Type**: This value MUST be zero.

**Global**: This value MUST be one.

**SiteID**: The **Site Collection Identifier** of the site collection. This parameter MUST NOT be NULL.

**UserID**: The **User Identifier** of the owner of the site group. This MUST be NULL when the site group owner is a site group.

**UserSID**: The **SystemId** of the owner of the site group. This MUST be NULL when the site group owner is a site group.

**UserName**: The user-friendly name of the owner of the site group. This MUST be NULL when the site group owner is a site group.

**UserLogin**: The login name of the owner of the site group. This MUST be NULL when the site group owner is a site group.

**UserEmail**: The e-mail address of the owner of the site group. This MUST be NULL when the site group owner is a site group.

**UserNotes**: The notes associated with the owner of the site group. This MUST be NULL when the site group owner is a site group.

**UserSiteAdmin**: A bit flag specifying whether the site group owner is a site collection administrator. When the site group owner is a site collection administrator, the **UserSiteAdmin** flag MUST be set to one. If the user or domain group owner of the site group is not a site collection administrator, the **UserSiteAdmin** flag MUST be set to zero. This MUST be NULL when the site group owner is a site group.

**UserDomainGroup**: A bit flag specifying whether the site group owner is a domain group. When the site group owner is a domain group, the flag MUST be set to one. When the site group owner is a user or domain group, the flag MUST be set to zero. This flag MUST be NULL when the site group owner is a site group.

**UserFlags**: Contains the **User Information Flags** value (section 2.2.12) describing the owner of the site group. This MUST be NULL when the site group owner is a site group.
**GroupID:** The **Site Group Identifier** of the owner of this site group. This MUST be NULL when the site group owner is a user or domain group.

**GroupName:** The user-friendly name of the owner of this site group. This MUST be NULL when the site group owner is a user or domain group.

**GroupDescription:** The description of the owner of this site group. This MUST be NULL when the site group owner is a user or domain group.

**GroupSiteID:** The **Site Collection Identifier** of the site collection containing the owner of this site group. This MUST be NULL when the site group owner is a user or domain group.

**GroupOwner:** The **User Identifier** or **Site Group Identifier** of the owner of the site group that owns this site group. This MUST be NULL when the site group owner is a user or domain group, because only an owner that is a site group can have a group owner.

**GroupOwnerIsUser:** A bit flag specifying whether the site group that owns this site group is in turn owned by a user or domain group, or by a site group. When the owner of the site group that owns this site group is a user or domain group, the **GroupOwnerIsUser** flag MUST be set to one. This flag MUST be set to zero when the site group owner is owned by a site group. This MUST be NULL when the site group owner is a user or domain group.

**DLAlias:** The e-mail distribution list address for the site group. This value MUST be NULL if the group (2) has no e-mail distribution list address.

**DLErrorMessage:** Contains the most recent error message returned by an asynchronous e-mail distribution list operation, if any.

**DLFlags:** Contains a bit field of status flags for the e-mail distribution list associated with this site group. This parameter MUST be one of the values listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>None.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Pending - If this bit is set, the e-mail distribution list associated with this site group currently has a pending asynchronous operation.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Dirty - If this bit is set, the e-mail distribution list associated with this group (2) is dirty. The e-mail distribution list is considered dirty when A) an asynchronous operation is pending, and B) the site group has been modified since the asynchronous operation was initiated. A dirty e-mail distribution list needs to have its status and membership synchronized as soon as the pending operation completes (whether it succeeds or fails).</td>
</tr>
<tr>
<td>0x00000004</td>
<td>PendingJobsIsRename - The type of the currently pending asynchronous e-mail distribution list operation, if any. When this bit is set, the pending operation is a Rename. When the bit is not set, it is a Create operation.</td>
</tr>
</tbody>
</table>

**DLJobId:** Contains the job identifier of the currently pending asynchronous e-mail distribution list operation, or zero if there is no pending operation.

**DLArchives:** An array of bytes containing a list of pairs of **List Identifiers** (section 2.2.1.1.5) defining additional lists, which are recipients of e-mail sent to the e-mail distribution list via the address in **DLAlias**. Each Identifier MUST be stored as a string, with commas separating the list's parent site **Document Identifier** (section 2.2.1.1.2) and the **List Identifier**, and with semicolons following each pair.

**RequestEmail:** The e-mail address used to send a request to join or depart a site group.
**Flags**: Contains the membership permissions bit field for the site group. This parameter MUST NOT be NULL and MUST be one of the values listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Allow anyone to view the membership of the site group.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Only allow members of the site group to view the membership.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Allow members of the site group to edit the membership of the site group.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>Allow users to request membership in this site group, and allow users to request to leave the site group. All requests MUST be sent to the e-mail address specified by RequestEmail.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>Automatically accept user requests to join or leave the site group. This bit MUST be set only when the bit 0x000000004 is also set.</td>
</tr>
</tbody>
</table>

### 2.2.6.6 AllSites Table

The **AllSites** table stores information about site collections. The table is defined using T-SQL syntax, as follows.

```sql
TABLE AllSites(
    FullUrl                        nvarchar(260) DEFAULT N'",
    Id                             uniqueidentifier,
    RootWebId                      uniqueidentifier,
    ClientTag                      smallint, DEFAULT 0 NOT NULL,
    NextUserOrGroupId              int,
    NextAppPrincipalId             int NOT NULL DEFAULT 1,
    OwnerID                        int,
    SecondaryContactID             int,
    Subscribed                     bit,
    TimeCreated                    datetime,
    Deleted                        bit,
    UsersCount                     int           DEFAULT ((1)),
    BWUsed                         bigint,
    DiskUsed                       bigint,
    SecondStageDiskUsed            bigint,
    QuotaTemplateID                smallint,
    DiskQuota                      bigint,
    UserQuota                      int,
    DiskWarning                    bigint,
    DiskWarned                     datetime,
    CurrentResourceUsage           float DEFAULT 0 NOT NULL,
    AverageResourceUsage           float DEFAULT 0 NOT NULL,
    ResourceUsageWarning           float DEFAULT 0 NOT NULL,
    ResourceUsageMaximum           float DEFAULT 0 NOT NULL,
    BitFlags                       int,
    SecurityVersion                bigint,
    DenyPermMask                   tPermMask DEFAULT 0 NOT NULL,
    CertificationDate              datetime,
    DeadWebNotifyCount             smallint,
    PortalURL                      nvarchar(260),
    PortalName                     nvarchar(255),
    LastContentChange              datetime DEFAULT (getutcdate()),
    LastSecurityChange             datetime DEFAULT (getutcdate()),
    AuditFlags                     int,
    InheritAuditFlags              int,
    UserInfoListId                 uniqueidentifier,
    UserIsActiveFieldRowOrdinal    int,
    UserIsActiveFieldColumnName    nvarchar(64),
    UserAccountDirectoryPath       nvarchar(512),
)
```

[MS-WSSFO3] - v20170620
Windows SharePoint Services (WSS): File Operations Database Communications Version 3 Protocol
Copyright © 2017 Microsoft Corporation
Release: June 20, 2017
HashKey                        binary(16),
DomainGroupMapVersion          bigint,
DomainGroupMapCacheVersion     bigint        DEFAULT {{-1}},
DomainGroupMapCache            varbinary(max),
HostHeader                     nvarchar(128),
SiteUrls                       varbinary(max),
SubscriptionId                 varbinary(16),
RbsCollectionId                int           DEFAULT 0 NOT NULL,
LastSMRequest                  datetime      DEFAULT NULL,
PlatformVersion             nvarchar(64) DEFAULT NULL,
UpgradeFlags                    int  DEFAULT 1 NOT NULL,
SourceSiteId                   uniqueidentifier SPARSE NULL,
ExpirationDate                 datetime      SPARSE NULL,
EvalSiteId                     uniqueidentifier SPARSE NULL,
UpgradeReminderDate            datetime,
SubscriptionName               nvarchar(48) DEFAULT NULL,
AppSiteDomainId                varchar(6) DEFAULT NULL,
NextAppWebDomainId             binary(4) DEFAULT NULL
);

FullUrl: The absolute URL of the site collection.

Id: The Site Collection Identifier of the site collection. Used only during upgrade.

RootWebId: The site identifier of the site (2) configured as the root site in the site collection.

ClientTag: A number that represents the state of the application files of the site (2).

NextUserOrGroupId: An integer value that is incremented when a new user or permission level is added to the site collection. Indicates the value of the next user or Site Group Identifier (section 2.2.1.1.10) to be used.

NextAppPrincipalId: An integer value that is incremented when a new app principal is added to the site collection. Indicates the value of the next app principal integer identifier in the site collection to be used.

OwnerID: The user identifier of the user who owns the site collection.

SecondaryContactID: The user identifier of the user who is the secondary contact of the site collection.

Subscribed: A bit set to 1 to indicate that the site collection has been subscribed for implementation-specific notifications.

TimeCreated: The date and time in UTC format when the site collection was created.

Deleted: A bit set to 1 indicates that a delete operation has been initiated on the site collection and it has been scheduled for deletion.

UsersCount: The number of users in the site collection.

BWUsed: The number of sites in the site collection actively used. Serves as an implementation-specific, usage-reporting feature.

DiskUsed: The size, in bytes, of disk space used to store content in the site collection.

SecondStageDiskUsed: The size, in bytes, of disk space used to store the second-stage trash bin items for the site collection.

QuotaTemplateID: The identifier of a quota template used to set the disk quota for the site collection.
**DiskQuota**: The maximum size, in bytes, of disk space that can be allocated by the site collection. A value of 0 indicates that no limit is set.

**UserQuota**: The maximum number of users that the site collection can contain. A value of 0 indicates that no limit is set.

**DiskWarning**: The size, in bytes, of disk space that can be allocated on the site collection before the disk warning email is sent. A value of 0 indicates that no warning email will be sent.

**DiskWarned**: A date and time value in UTC format set to the last time a warning was sent to the site collection owner about disk usage, if any.

**CurrentResourceUsage**: A float that specifies the current resource usage of the site collection up to the current date.

**AverageResourceUsage**: A float that specifies the average resource usage of the site collection over the configured period of days.

**ResourceUsageWarning**: A float that specifies the current resource usage warning setting. If the current resource usage exceeds it, a warning notification will be sent.

**ResourceUsageMaximum**: A float that specifies the maximum resource usage of the site collection.

**BitFlags**: A Site Collection Flags (section 2.2.2.9) value describing the site collection.

**SecurityVersion**: A version number that is incremented when changes are made to the site collection's permissions.

**DenyPermMask**: A WSS Rights Mask (section 2.2.2.15) specifying the rights to deny to the current user.

**CertificationDate**: The date and time, in UTC format, when the site collection was last confirmed by its owner as being used.

**DeadWebNotifyCount**: The number of times that a notification was sent to the site collection owner that the site collection will be deleted if the owner does not certify it as being used. (See the CertificationDate column.)

**PortalURL**: The URL of an external Web site designated as the portal of the site (1). This is an implementation-specific navigation links feature.

**PortalName**: The name of the external Web site referenced in the PortalURL column. Used for display in implementation-specific navigation features.

**LastContentChange**: The date and time value, in UTC format, that the content of the site collection was last changed.

**LastSecurityChange**: The date and time value, in UTC format, when the permissions on the site collection were last changed.

**AuditFlags**: An Audit Flags (section 2.2.2.1) value specifying the operations to be audited that are set directly on the site collection.

**InheritAuditFlags**: An Audit Flags (section 2.2.2.1) value specifying the operations to be audited on the specified object that are inherited.

**UserInfoListId**: The List Identifier (section 2.2.1.1.5) of a list holding user information for the site collection.

**UserIsActiveFieldRowOrdinal**: The ordinal of the column in the list identified by UserInfoListId that tracks whether a user is an active user in the site collection.
**UserIsActiveFieldColumnName**: The name of the column in the list identified by **UserInfoListId** that tracks whether a user is an active user in the site collection.

**UserAccountDirectoryPath**: A provider-specific user account path. The value is used for validation of added user accounts if the site collection flags bit 0x00080000 is set in the **BitFlags** column.

**HashKey**: A **hash** key associated with the site collection.

**DomainGroupMapVersion**: The version of the **domain group** map.

**DomainGroupMapCacheVersion**: The version of the domain group map cache.

**DomainGroupMapCache**: A binary serialization of the domain group map cache, containing a mapping of external groups to the permission levels of which they are **members**.

**HostHeader**: When used in **host header** mode, this contains the host header string associated with the site collection's **Web application**.

**SiteUrls**: A string that contains a semi-colon separated list of encoded URLs of the site collection. Each URL is encoded as a comma separated set of properties:

<UrlZone:1 char encoding **tinyint** as ascii digit><comma><Scheme: string, its length is less or equal than 5><comma><HostHeader><comma><AppSiteDomainId>

**SubscriptionId**: The **GUID** of the implementation-specific subscription feature for the site collection.

**RbsCollectionId**: The identifier of the remote **BLOB** storage collection associated with the site collection, or zero if remote BLOB storage is not configured for this database.

**LastSMRequest**: The date and time value, in UTC format, when the storage metrics information for the site collection was last requested.

**PlatformVersion**: The string that represents the current version of the site collection from the platform compatibility perspective.

**UpgradeFlags**: The site collection upgrade flags (see section **2.2.10**) for this site collection.

**SourceSiteId**: The **site collection identifier** of the site collection that was used to create the upgrade evaluation site collection. A non-NULL value for this column implies that the site collection is an upgrade evaluation site collection.

**ExpirationDate**: The date and time value in UTC format, after which the upgrade evaluation site collection will be automatically deleted.

**EvalSiteId**: The site collection identifier of the upgrade evaluation site collection that was created using the site collection.

**UpgradeReminderDate**: The upgrade reminder date of the site collection.

**SubscriptionName**: The name corresponding to the **site subscription identifier** of the implementation-specific subscription for the requested site collection.

**AppSiteDomainId**: The **app site domain identifier**.

**NextAppWebDomainId**: The next **app web domain identifier** that will be used for the next subsite created within the site collection.
2.2.6.7 Sites View

The Sites View reflects all rows in the table AllSites (section 2.2.6.6) that are not scheduled for deletion (that is, all table rows where Deleted is equal to 0). Sites View records information about site collections.

See AllSites for the T-SQL syntax description of the structure of this view.

2.2.6.8 UserData View

The UserData View reflects all list items in the AllUserData table (section 2.2.6.2) that are current versions and not marked as deleted (that is, all table rows where tp_IsCurrentVersion is 1, tp_CalculatedVersion is 0, and tp_DeleteTransactionID is 0x).

See AllUserData for the T-SQL syntax description of the structure of this view. Columns defined with the keyword SPARSE do not appear in this view.

2.2.6.9 UserDataVersioned View

The UserDataVersioned View reflects all list items in the AllUserData table (section 2.2.6.2) that are not deleted (that is, all table rows where tp_DeleteTransactionID is 0x).

See AllUserData for the T-SQL syntax description of the structure of this view.

2.2.6.10 UserInfo Table

The UserInfo Table stores descriptive properties and security information about principals with access to a site collection. The UserInfo Table is defined using T-SQL syntax, as follows.

```sql
TABLE UserInfo (
    tp_SiteID uniqueidentifier NOT NULL,
    tp_Id int NOT NULL,
    tp_DomainGroup bit NOT NULL,
    tp_SystemID varbinary(512) NOT NULL,
    tp_Deleted int NOT NULL,
    tp_SiteAdmin bit NOT NULL,
    tp_IsActive bit NOT NULL DEFAULT (1),
    tp_Login nvarchar(255) NOT NULL,
    tp_Title nvarchar(255) NOT NULL,
    tp_Email nvarchar(255) NOT NULL,
    tp_Notes nvarchar(1023) NOT NULL,
    tp_Token varbinary(max) NULL,
    tp_ExternalToken varbinary(max) NULL,
    tp_ExternalTokenLastUpdated datetime NULL,
    tp_Locale int NULL,
    tp_CalendarType smallint NULL,
    tp_AdjustHijriDays smallint NULL,
    tp_TimeZone smallint NULL,
    tp_Time24 bit NULL,
    tp_AltCalendarType tinyint NULL,
    tp_CalendarViewOptions tinyint NULL,
    tp_WorkDays smallint NULL,
    tp_WorkDayStartHour smallint NULL,
    tp_WorkDayEndHour smallint NULL,
    tp_Mobile nvarchar(127) NULL,
    tp_MUILanguages varchar(64) NULL,
    tp_ContentLanguages varchar(64) NULL,
    tp_Flags int NOT NULL DEFAULT (0),
);```
tp_SiteID: The Site Collection Identifier (section 2.2.1.9) of the site collection associated with the principal (1).

tp_Id: The User Identifier (section 2.2.1.13) of the principal, which MUST be unique within both this table and the group identifiers for permission levels. Certain User Identifiers are predefined, such as the system account (1073741823), the site collection owner (1), and the secondary site collection contact (2). User Identifier values of 0 and -1 MUST NOT be used in the table.

tp_DomainGroup: A bit set to 1 if the principal is a domain group; otherwise, 0, specifying that the principal is a user.

tp_SystemID: The SystemId of the principal.

tp_Deleted: A value specifying whether the principal is marked as deleted. If tp_Deleted is 0, the principal is not deleted. If tp_Deleted is not 0, the principal is marked as deleted, and the value is the User Identifier that was associated with this principal. The deleted state can be used for user information, rather than dropping entries from the table, to preserve list item ownership information. A user or domain group with the tp_Deleted value set to nonzero can be restored by setting the tp_Deleted value to 0 and updating other fields as necessary.

tp_SiteAdmin: A bit set to 1 if the principal is a site collection administrator for the site collection.

tp_IsActive: A bit set to 1 if the principal is an active user in the site collection.

tp_Login: The login name for the principal. This value MUST NOT be empty.

tp_Title: The display name for the principal. This value MUST NOT be empty.

tp_Email: The e-mail address for the principal. This value can be empty.

tp_Notes: A descriptive text about the principal. This value can be empty.

tp_Token: A WSS User Token (section 2.2.3.10) value specifying the permission level membership of the principal. This can be NULL, indicating that the principal is not a member of a permission level.

tp_ExternalToken: An External Group Token (section 2.2.3.2) value encoding information on external group membership derived from an external authentication role provider. This value can be NULL, indicating that this user has never visited any site in the site collection. If this value is NULL, the value in tp_ExternalToken LastUpdated MUST also be NULL.

tp_ExternalTokenLastUpdated: A datet ime containing a time stamp in UTC format specifying the time when the value in tp_ExternalToken for the principal was last updated. This can be NULL if the value in tp_ExternalToken has never been updated.

tp_Locale: An LCID specifying the preferred locale value to be used when displaying messages in the front-end Web server for the principal. This can be NULL, specifying that the system or site default locale can be used instead.

tp_CalendarType: The Calendar Type to be used when processing date values for the principal. This can be NULL, specifying that the site default Calendar Type can be used instead.

tp_AdjustHijriDays: If the tp_CalendarType value is "6", this specifies the number of days to extend or reduce the current month in Hijri calendars for the principal. Otherwise, this value MUST be NULL and MUST be ignored.

tp_TimeZone: The Time Zone Identifier for the time zone to be used when displaying time values for the principal. This can be NULL, specifying that the system or site default time zone can be used instead.

tp_Time24: A bit flag specifying whether a 24-hour time format can be used when displaying time values to the principal. If tp_Time24 is 1, the 24-hour time format will be used; otherwise, the 12-
hour time format will be used. This can be NULL, specifying that the system or site default can be used instead.

**tp_AltCalendarType**: The Calendar Type of an alternate calendar to be used when displaying calendars in views for the principal. This can be NULL, specifying that the **tp_CalendarType** value can be used instead.

**tp_CalendarViewOptions**: A Calendar View Options Type that specifies the calendar options for the principal. This can be NULL, specifying that the site default calendar view options can be used instead.

**tp_WorkDays**: A set of Workdays Flags that specify the weekdays defined as the workweek for the principal. This can be NULL, specifying that the site default workdays value can be used instead.

**tp_WorkDayStartHour**: The start time of the workday in minutes after midnight for the principal. This can be NULL, specifying that the site default workday start hour can be used instead.

**tp_WorkDayEndHour**: The end time of the workday in minutes after midnight for the principal. This can be NULL, specifying that the site default workday end hour can be used instead.

**tp_Mobile**: The mobile phone number of the security principal.

**tp_MUILanguages**: A string that contains the distinct culture(s) for the user's preferred display language(s), separated by semicolon. This string can be NULL.

**tp_ContentLanguages**: A string that contains the distinct culture(s) for the user's preferred content language(s), separated by semicolon. This string can be NULL.

**tp_Flags**: A 4-byte integer bit mask determining the user's options. For more details, see section 2.2.2.12.

### 2.2.6.11 WebMembers

The **WebMembers** table stores information used for determining the membership of users in sites. This table is defined using T-SQL syntax, as follows.

```sql
TABLE WebMembers ( 
    SiteId                        uniqueidentifier NOT NULL, 
    WebId                         uniqueidentifier NOT NULL, 
    UserId                        int NOT NULL
);
```

**SiteId**: The site collection identifier of the site collection that contains this site (2).

**WebId**: The site identifier of the site (2) associated with the user.

**UserId**: The user identifier of the user associated with the site (2).

### 2.2.6.12 Versions

The **Versions Table** stores information used for object versioning during upgrade operations. This Table is defined using T-SQL syntax, as follows.

```sql
TABLE Versions ( 
    VersionId                     uniqueidentifier NOT NULL, 
    Version                       nvarchar(64) NOT NULL, 
    Id                            int IDENTITY(1,1) NOT NULL, 
    UserName                      nvarchar(255) NULL, 
    TimeStamp                     datetime NULL,
);
```
VersionId: A GUID that identifies the version.

Version: The actual version string in the format ":<major>.<minor>.<build>.<iteration>" (for example, "3.0.106.0").

Id: An integer value that is an SQL identity column.

UserName: A string containing the user name adding or updating the version.

TimeStamp: A time stamp in UTC format indicating when the version was added or updated.

FinalizeTimeStamp: Unused.

Mode: Unused.

ModeStack: Unused.

Updates: An integer value tracking the number of updates applied to a particular version. This value MUST be initialized to 1 when inserted and incremented each time the version is updated.

Notes: A string containing optional notes associated with the version.

2.2.7 XML Structures

No common XML Structures are defined in this protocol.

2.2.7.1 Namespaces

This specification does not define any common XML schema namespaces.

2.2.7.2 Simple Types

2.2.7.2.1 FALSE_Case_Inensitive_Else_Anything

This type is used to specify a Boolean value.

```xml
<xs:simpleType name="FALSE_Case_Inensitive_Else_Anything">
  <xs:restriction base="xs:string">
    <xs:pattern value="[Ff][Aa][Ll][Ss][Ee]|.*" />
  </xs:restriction>
</xs:simpleType>
```

2.2.7.2.2 FieldInternalType

The FieldInternalType type is used to specify a field internal type.

Note: In the following XML, the value attribute is broken into multiple lines only for readability.

```xml
<xs:simpleType name="FieldInternalType">
```
While the XSD pattern is normative, the values are:

- Integer
- Text
- Note
- datetime
- Counter
- Choice
- Lookup
- Boolean
- Number
- Currency
- Url
- Computed
- Threading
- GUID
- MultiChoice
- GridChoice
- Calculated
- File
- Attachments
- User
- Recurrence
- CrossProjectLink
- ModStat
- Error
- ContentTypeId
- PageSeparator
- ThreadIndex
- WorkflowStatus
- AllDayEvent
- WorkFlowEventType

### 2.2.7.2.3 FieldAggregationAttribute

The **FieldAggregationAttribute** type is used to specify how values are promoted from an XML or site template document.

```xml
<xs:simpleType name="FieldAggregationAttribute">
  <xs:restriction base="xs:string">
    <xs:enumeration value="average"/>
    <xs:enumeration value="count"/>
    <xs:enumeration value="first"/>
    <xs:enumeration value="last"/>
    <xs:enumeration value="max"/>
    <xs:enumeration value="merge"/>
    <xs:enumeration value="min"/>
    <xs:enumeration value="plaintext"/>
    <xs:enumeration value="signature"/>
    <xs:enumeration value="sum"/>
  </xs:restriction>
</xs:simpleType>
```

The meanings of the values are specified in the following table. In the table, a matching node means an element or an attribute that matches a specified XPath query. The value of an element node is the concatenation of all character data directly contained by the element.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>average</td>
<td>The average of the matching nodes, whose values can be interpreted as a floating point numbers. Empty string if no values match.</td>
</tr>
<tr>
<td>count</td>
<td>Count of the matching nodes.</td>
</tr>
<tr>
<td>first</td>
<td>The value of the first matching node.</td>
</tr>
<tr>
<td>last</td>
<td>The value of the last matching node.</td>
</tr>
<tr>
<td>max</td>
<td>The value of the largest matching node interpreted as a floating point number.</td>
</tr>
<tr>
<td>merge</td>
<td>The value of the nodes, in order, delimited by a newline character.</td>
</tr>
<tr>
<td>min</td>
<td>The value of the smallest matching node interpreted as a floating point number.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>plaintext</td>
<td>If no nodes match, or if the first matching node is not an element, an empty string. Otherwise, renders the first matching node and its child elements with only character data shown.</td>
</tr>
<tr>
<td>signature</td>
<td>TRUE if the first matching node is an element and has child elements. FALSE if the first matching node is an element with no child elements. Otherwise, an empty string.</td>
</tr>
<tr>
<td>sum</td>
<td>The sum of the nodes that match and whose values can be interpreted as a floating point numbers.</td>
</tr>
</tbody>
</table>

2.2.7.2.4 FieldRefType

The FieldRefType type is used to specify the type of relationship given by a field reference definition within the field.

```xml
<xs:simpleType name="FieldRefType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Recurrence" />
    <xs:enumeration value="EventType" />
    <xs:enumeration value="UID" />
    <xs:enumeration value="RecurrenceId" />
    <xs:enumeration value="EventCancel" />
    <xs:enumeration value="StartDate" />
    <xs:enumeration value="EndDate" />
    <xs:enumeration value="RecurData" />
    <xs:enumeration value="Duration" />
    <xs:enumeration value="TimeZone" />
    <xs:enumeration value="XMLTZone" />
    <xs:enumeration value="CPLink" />
    <xs:enumeration value="LinkURL" />
    <xs:enumeration value="MasterSeriesItemID" />
    <xs:enumeration value="AllDayEvent" />
  </xs:restriction>
</xs:simpleType>
```

The meanings of the values are specified in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recurrence</td>
<td>Specifies the recurrence field.</td>
</tr>
<tr>
<td>EventType</td>
<td>Specifies the field that contains the type of the event.</td>
</tr>
<tr>
<td>UID</td>
<td>Specifies the field that contains the identifier for the event.</td>
</tr>
<tr>
<td>RecurrenceId</td>
<td>Specifies the field that contains the recurrence identifier for the event.</td>
</tr>
<tr>
<td>EventCancel</td>
<td>Specifies the field that identifies whether the event has been canceled.</td>
</tr>
<tr>
<td>StartDate</td>
<td>Specifies the field that contains the start date for the event.</td>
</tr>
<tr>
<td>EndDate</td>
<td>Specifies the field that contains the end date for the event.</td>
</tr>
<tr>
<td>RecurData</td>
<td>Specifies the field that contains recurrence information for the event.</td>
</tr>
<tr>
<td>Duration</td>
<td>Specifies the field that contains the duration of the event.</td>
</tr>
<tr>
<td>TimeZone</td>
<td>Specifies a reference to the effective time zone for the event.</td>
</tr>
<tr>
<td>XMLTZone</td>
<td>Specifies a reference to XML that describes the effective time zone for the event.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CPLink</td>
<td>Specifies a reference to another web site.</td>
</tr>
<tr>
<td>LinkURL</td>
<td>Specifies the field that contains a link to another web site.</td>
</tr>
<tr>
<td>MasterSeriesItemID</td>
<td>Specifies the field that contains the ID of the item that defines the recurring event.</td>
</tr>
<tr>
<td>AllDayEvent</td>
<td>Specifies the field that identifies whether the event is an all-day event.</td>
</tr>
</tbody>
</table>

### 2.2.7.2.5 FieldRichTextMode

The **FieldRichTextMode** type is used to specify a formatting mode for a rich text field.

```xml
<xs:simpleType name="FieldRichTextMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Compatible"/>
    <xs:enumeration value="FullHtml"/>
    <xs:enumeration value="HtmlAsXml"/>
    <xs:enumeration value="ThemeHtml"/>
  </xs:restriction>
</xs:simpleType>
```

The meanings of the values are specified in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compatible</td>
<td>Specifies support for basic formatting, such as bold formatting.</td>
</tr>
<tr>
<td>FullHtml</td>
<td>Specifies support for more advanced formatting options, such as pictures, tables, and hyperlinks.</td>
</tr>
<tr>
<td>HtmlAsXml</td>
<td>Specifies support for HTML that is well-formed XML.</td>
</tr>
<tr>
<td>ThemeHtml</td>
<td>Specifies support for formatting with in-line style specifications. See [HTML], section 14.2.2.</td>
</tr>
</tbody>
</table>

### 2.2.7.2.6 IMEMode

The **IMEMode** type is used to specify a bias for an **Input Method Editor (IME)**.

```xml
<xs:simpleType name="IMEMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="inactive" />  
    <xs:enumeration value="auto" />      
    <xs:enumeration value="active" />    
    <xs:enumeration value="disabled" />  
  </xs:restriction>
</xs:simpleType>
```

The reader MUST accept any value. The writer SHOULD write this attribute as one of the valid values for the `ime-mode` attribute (as described in [CSS3UI]) to be applied to an `<input>` tag (as specified in [HTML], section 17.4).

### 2.2.7.2.7 IntPositive
This type is used to specify a positive integer.

```xml
<sxs:simpleType name="IntPositive">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1" />
  </xs:restriction>
</sxs:simpleType>
```

### 2.2.7.2.8 JoinType

The **JoinType** type specifies how to handle lookup fields for which the target of the lookup does not exist.

```xml
<sxs:simpleType name="JoinType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="INNER" />
    <xs:enumeration value="LEFT OUTER" />
  </xs:restriction>
</sxs:simpleType>
```

The meanings of the values are specified in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INNER</td>
<td>Items whose target does not exist are omitted.</td>
</tr>
<tr>
<td>LEFT OUTER</td>
<td>Items whose target does not exist are included.</td>
</tr>
</tbody>
</table>

### 2.2.7.2.9 TextDirection

The **TextDirection** type is used to specify the preferred direction of displaying text.

```xml
<sxs:simpleType name="TextDirection">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ltr" />  
    <xs:enumeration value="rtl" />  
    <xs:enumeration value="none" /> 
    <xs:enumeration value="None" /> 
  </xs:restriction>
</sxs:simpleType>
```

The meanings of the values are specified in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ltr</td>
<td>Specifies a left-to-right text direction.</td>
</tr>
<tr>
<td>rtl</td>
<td>Specifies a right-to-left text direction.</td>
</tr>
<tr>
<td>none</td>
<td>Specifies no hint for a text direction, and that the direction will follow the context of the page.</td>
</tr>
<tr>
<td>None</td>
<td>Same as none.</td>
</tr>
</tbody>
</table>

### 2.2.7.2.10 TRUE_If_Present
This type is used to specify a Boolean value.

```xml
<xs:simpleType name="TRUE_If_Present">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

### 2.2.7.2.11 TRUEFALSE

This type is used to specify a Boolean value.

```xml
<xs:simpleType name="TRUEFALSE">
  <xs:restriction base="xs:string">
    <xs:pattern value="\[Tt\][Rr]\[Uu]\[Ee]\|\[Ff]\[Aa]\[Ll]\[Ss]\[Ee]\="/>
  </xs:restriction>
</xs:simpleType>
```

### 2.2.7.2.12 TRUE_Case_Sensitive_Else.Anything

This type is used to specify a Boolean value.

```xml
<xs:simpleType name="TRUE_Case_Sensitive_Else.Anything">
  <xs:restriction base="xs:string">
    <xs:pattern value="TRUE|.*"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.2.7.2.13 UniqueIdentifierWithBracesUppercase

This type is used to specify a GUID.

```xml
<xs:simpleType name="UniqueIdentifierWithBracesUppercase">
  <xs:restriction base="xs:string">
    <xs:pattern value="\{\[0-9A-F\]{8}\}-\{0-9A-F\}{4}\-\{0-9A-F\}{4}\-\{0-9A-F\}{4}\-\{0-9A-F\}{12}\}"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.2.7.2.14 UniqueIdentifierWithOrWithoutBraces

This type is used to specify a GUID.

```xml
<xs:simpleType name="UniqueIdentifierWithOrWithoutBraces">
  <xs:restriction base="xs:string">
    <xs:pattern value="\{\[0-9a-fA-F\]{8}\}-\{0-9a-fA-F\}{4}\-\{0-9a-fA-F\}{4}\-\{0-9a-fA-F\}{4}\-\{0-9a-fA-F\}{12}\}"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.2.7.2.15 RelationshipDeleteBehaviorAttribute

The `RelationshipDeleteBehaviorAttribute` describes the relationship between a list item and the target list item of a `Lookup` field.

```xml
<xs:simpleType name="RelationshipDeleteBehaviorAttribute">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Cascade"/>
  </xs:restriction>
</xs:simpleType>
```
The meanings of the values are specified in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascade</td>
<td>Specifies that when a list item in the target list is deleted, all related items in the list that contains the Lookup field are also deleted.</td>
</tr>
<tr>
<td>Restrict</td>
<td>Specifies that deleting a list item in the target list is prevented if there are any related items in the list that contains the Lookup field.</td>
</tr>
<tr>
<td>None</td>
<td>Specifies no restrictions.</td>
</tr>
</tbody>
</table>

2.2.7.3 Complex Types

2.2.7.3.1 CHOICEDEFINITION

The CHOICEDEFINITION type contains a choice for a Choice or MultiChoice field.

2.2.7.3.1.1 Schema

```xml
<xs:complexType name="CHOICEDEFINITION" mixed="true">
  <xs:attribute name="JumpTo" type="xs:string" />
</xs:complexType>
```

2.2.7.3.1.2 Attributes

JumpTo: Specifies the Name of the next field in a survey list to display when this choice is selected.

2.2.7.3.1.3 Child Elements

<Content>: Specifies a value for a choice in a Choice or MultiChoice field.

2.2.7.3.2 CHOICEDEFINITIONS

The CHOICEDEFINITIONS type contains a collection of choices for a Choice or MultiChoice field.

2.2.7.3.2.1 Schema

```xml
<xs:complexType name="CHOICEDEFINITIONS">
  <xs:sequence>
    <xs:element name="CHOICE" type="CHOICEDEFINITION" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

2.2.7.3.2.2 Attributes

None.
2.2.7.3.2 Child Elements

**CHOICE**: Specifies a choice in a **Choice** or **MultiChoice** field.

2.2.7.3 FieldDefinition

A field definition describes the structure and format of a field that is used within a list or content type.

2.2.7.3.1 Schema

```xml
<xs:complexType name="FieldDefinition" mixed="true">
  <xs:all>
    <xs:element name="CHOICES" type="CHOICEDEFINITIONS" minOccurs="0" maxOccurs="1" />
    <xs:element name="Customization" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:any minOccurs="1" maxOccurs="1" namespace="##any" processContents="skip" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Default" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="DefaultFormula" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="DisplayBidiPattern" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##any" processContents="skip" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="DisplayPattern" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##any" processContents="skip" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="FieldRefs" minOccurs="0" maxOccurs="1">
      <xs:complexType mixed="true">
        <xs:sequence>
          <xs:element name="FieldRef" type="FieldRefDefinitionField" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Formula" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="FormulaDisplayNames" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="MAPPINGS" type="MAPPINGDEFINITIONS" minOccurs="0" maxOccurs="1" />
    <xs:element name="ParserRefs" type="FieldParserRefs" minOccurs="0" maxOccurs="1" />
    <xs:element name="Validation" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##any" processContents="skip" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:all>
  <xs:attribute name="Aggregation" type="FieldAggregationAttribute" default="first" />
  <xs:attribute name="aggregation" type="xs:string" />
  <xs:attribute name="AllowDeletion" type="TRUEFALSE" default="TRUE" />
  <xs:attribute name="AllowHyperlink" type="TRUEFALSE" default="FALSE" />
  <xs:attribute name="AllowMultiVote" type="TRUEFALSE" default="FALSE" />
</xs:complexType>
```
<xs:attribute name="AppendOnly" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="AuthoringInfo" type="xs:string" default="" />
<xs:attribute name="BaseRenderingType" type="FieldInternalType" />
<xs:attribute name="CalType" type="xs:int" />
<xs:attribute name="CalendarType" type="xs:int" />
<xs:attribute name="CAMLRendering" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="CanToggleHidden" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="CountRelated" type="TRUE_IF_Present" default="FALSE" />
<xs:attribute name="ClassInfo" type="xs:string" default="" />
<xs:attribute name="ColName" type="xs:string" />
<xs:attribute name="Commas" type="TRUEFALSE" />
<xs:attribute name="Customization" type="xs:string" />
<xs:attribute name="Decimals" type="xs:int" default="-1" />
<xs:attribute name="DefaultListField" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="DefaultURLDesc" type="xs:string" />
<xs:attribute name="Description" type="xs:string" />
<xs:attribute name="DisplayImage" type="xs:string" />
<xs:attribute name="DisplaySize" type="xs:int" />
<xs:attribute name="Div" type="xs:string" default="1.0" />
<xs:attribute name="EnableLookup" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="EnforceUniqueValues" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ExceptionImage" type="xs:string" />
<xs:attribute name="FieldRef" type="xs:string" />
<xs:attribute name="FillInChoice" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="Filterable" type="TRUEFALSE" default="TRUE" />
<xs:attribute name="FilterableNoRecurrence" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ForcePromoteDemote" type="TRUE_IF_Present" />
<xs:attribute name="Format" type="xs:string" />
<xs:attribute name="FromBaseType" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="GridEndNum" type="xs:int" />
<xs:attribute name="GridNATxt" type="xs:string" default="" />
<xs:attribute name="GridStartNum" type="IntPositive" />
<xs:attribute name="GridTxtRng1" type="xs:string" default="" />
<xs:attribute name="GridTxtRng2" type="xs:string" default="" />
<xs:attribute name="GridTxtRng3" type="xs:string" default="" />
<xs:attribute name="Group" type="xs:string" />
<xs:attribute name="HeaderImage" type="xs:string" />
<xs:attribute name="Height" type="xs:int" />
<xs:attribute name="Hidden" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ID" type="UniqueIdentifierWithOrWithoutBraces" />
<xs:attribute name="ID" type="xs:string" />
<xs:attribute name="IMEMode" type="IMEMode" />
<xs:attribute name="Indexed" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ImnHeader" type="xs:string" />
<xs:attribute name="IsRelationship" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="JoinColName" type="xs:string" default="tp_ID" />
<xs:attribute name="JoinRowOrdinal" type="xs:int" fixed="0" />
<xs:attribute name="JoinType" type="JoinType" default="LEFT OUTER" />
<xs:attribute name="JumpTo" type="xs:string" />
<xs:attribute name="JumpToFillInChoice" type="xs:string" />
<xs:attribute name="JumpToNo" type="xs:string" />
<xs:attribute name="JumpToYes" type="xs:string" />
<xs:attribute name="LCID" type="xs:int" />
<xs:attribute name="LinkToItem" type="TRUE_Case_Sensitive_Else Anything" />
<xs:attribute name="LinkToItemAllowed" type="xs:string" />
<xs:attribute name="List" type="xs:string" />
<xs:attribute name="ListItemMenu" type="TRUE_Case_Sensitive_Else Anything" />
<xs:attribute name="ListItemMenuAllowed" type="xs:string" />
<xs:attribute name="Max" type="xs:float" />

194 / 484
2.2.7.3.3.2 Attributes

Aggregation: For fields with the Node or node attribute, a reader MUST use this attribute to control promotion from XML or site template files. For other fields, a reader MUST ignore this attribute.

aggregation: For fields whose Type attribute maps to the Note field internal type, the reader can permit a filtering user interface if this attribute contains the value merge (with case-insensitive comparison).<3> Otherwise a reader MUST ignore this attribute.

A writer SHOULD NOT include this attribute.

AllowDeletion: If this attribute is FALSE, then this is used to specify that a server MUST NOT permit the field to be removed from the schema of a list.

AllowHyperlink: For fields whose Type maps to the Note field internal type, and for which RichText is TRUE and RichTextMode is Compatible, a reader can use this attribute to determine if the editing UI allows insertion of hyperlinks. If this attribute is TRUE, a server or client presenting an editing user interface for this field MUST include the insert hyperlink command. Otherwise, the editing user interface MUST NOT include the insert hyperlink command.

AllowMultiVote: The reader MUST ignore this attribute.

AppendOnly: If TRUE, for fields whose Type maps to the Note field internal type, a client or server which presents a way of editing the value for this field MUST do so in a way that ensures preservation of old data in the field.

AuthoringInfo: Text describing the field to schema authors. A client or server presenting a schema-editing UI can display the text in this attribute.

BaseRenderingType: Field internal type that the server uses to render a field. If not present, the Type attribute does not map to an internal type in a way suitable for rendering.

The server MUST derive the value of this attribute from the Type attribute, overriding the client-specified value.

BaseType: Allows a schema author to override the internal storage format for choice fields. A reader MUST ignore this attribute unless the field’s internal type is Choice, MultiChoice, or GridChoice. A writer MUST NOT set this attribute’s value to Choice, MultiChoice, or GridChoice.

CalType: Associates a calendar type with a field. A reader SHOULD take this value into consideration when rendering the field or performing date calculations on it. If this attribute is not present, a reader MUST use the CalendarType attribute. If that is also not present, it SHOULD fall back to the user’s preference, and if that is not specified, to the calendar type for the site (2).

CalendarType: This is equivalent to CalType. A reader MUST use the value of CalType if present, but if that attribute is not present, it MUST use this attribute for the same purpose. A writer SHOULD omit this attribute, using CalType instead where it is needed.

Calculated: The reader MUST ignore this attribute.

CAMLRendering: If TRUE, a field MUST be rendered using its DisplayPattern child element in a list view.

CanToggleHidden: If TRUE, a server MUST permit the user to change the value of the Hidden attribute. Otherwise, a server MUST prevent the Hidden attribute from changing.
**CountRelated:** If TRUE, the internal type of the field MUST be **Lookup**, the **List** and **FieldRef** attributes of the field MUST refer to a field of internal type **Lookup**, and for the referred field, its **List** attribute MUST refer to the list of which the source field is a member. A writer MUST NOT include this attribute if the **Mult** attribute is present. A reader MUST ignore the attribute if the field internal type of the field is not **Lookup**.

**ClassInfo:** A writer SHOULD use values Menu or Icon, or it SHOULD omit the attribute. When rendering the value of the field in a view, a server SHOULD use this to format the field for a menu or an icon. The server MUST ignore values other than Menu or Icon and MUST fall back to the default behavior for other values.

**ColName:** A server SHOULD use this attribute to describe the physical storage location for the field. A client MUST ignore this attribute and a server MUST pick its own value for this attribute when accepting **CAML** from a client.

**ColName2:** This has the same restrictions and semantics as the **ColName** attribute except it describes a secondary physical storage location.

**Commas:** A writer SHOULD NOT use this attribute; a reader MUST ignore this attribute.

**Customization:** A writer SHOULD NOT use this attribute; a reader MUST ignore this attribute.

**Decimals:** Number of digits to render in the fractional part of a number. A writer SHOULD only use this attribute on fields whose field internal type is **Number** or **Calculated**. The reader MUST ignore this attribute on files with other internal types. A writer SHOULD omit this attribute rather than write the default value of "-1".

When rendering a numeric field, a server MUST use this attribute to determine the number of decimal places to render. For the special value of -1, trailing zeros MUST be omitted.

**DefaultListField:** A server or client can determine whether to request field values for a document or list item after a user action such as uploading a new document when editable fields are present on the list. When this attribute is set to TRUE, the server or client can ignore the presence of this field for the purposes of that determination.

**DefaultURLDesc:** For a field whose field internal type is **URL**, the value to use for the description part when the user does not provide a description. The default is to use the URL part of the field. If the field’s internal type is not **URL**, a reader MUST ignore this attribute.

**Description:** Textual description of the field to be displayed in the user interface for schema authoring.

**Direction:** Specifies that HTML field rendering will be done in a left-to-right, right-to-left, or in the ambient context of the surrounding page.

**Dir:** A writer SHOULD NOT use this attribute; a reader MUST ignore this attribute.

**DisplaceOnUpgrade:** When TRUE, indicates that a server depends on the name of a field. The client MUST NOT set this attribute.

**DisplayImage:** When specified, this attribute supplies the URL, relative to the server’s "/_layouts/images" folder, to render when displaying the field. A reader MUST ignore this attribute except for **CrossProjectLink** fields and **Recurrence** fields.

If not specified, a blank placeholder image is used instead.

**DisplayName:** Text to display in the user interface when referring to the field.

**DisplayNameSrcField:** When present, specifies the name of another field on the list to which the **DisplayName** of the current field is synchronized.
**DisplaySize:** Number of columns to offer when displaying the field.

**Div:** In fields whose field internal type is **Integer, Number, or Calculated**, a reader MUST interpret this as a floating point number by which the value is divided at render time.

In fields of other internal types, a reader MUST ignore this attribute and a writer MUST NOT include this attribute.

**EnableLookup:** Specifies whether the user interface for creating a **Lookup** field lists a **Computed** field as a possible target for the lookup. Unless the target field type is **Computed**, a writer SHOULD NOT include this attribute and a reader MUST ignore it.

**EnforceUniqueValues:** If this attribute is TRUE, then the server MUST NOT permit any list item to have the same value for the field as any other list item in the list, unless the value of the field is NULL.

**ExceptionImage:** Specifies an image to display in place of **DisplayImage** for recurrence fields for items that are exceptions to the recurrence. A writer MUST include this attribute for recurrence fields if **DisplayImage** is also included and MUST NOT include it otherwise.

**FieldRef:** When present on a **Lookup** field, specifies the **Name** of another field on the list from which to obtain the local value for the lookup. **Lookup** fields for which this attribute is not specified supply their own storage for the local value.

**FillInChoice:** Whether a **form** generated to let the user edit a **Choice** field or **MultiChoice** field allows values other than those listed in the **CHOICES** child element.

A reader MUST ignore this attribute except for **Choice** fields and **MultiChoice** fields.

**Filterable:** Whether items can be omitted or included from the results of a query based on the value of the field. True if the field can be filtered; false otherwise.

**FilterableNoRecurrence:** When TRUE, and the **Filterable** attribute is FALSE, specifies that items can be included or omitted from the results of a query in views that do not expand recurring **events**.

**ForcedDisplay:** If present, specifies a value for the field to display in place of the field's real value.

**Format:** Specifies how to render the field. The interpretation of this attribute and the range of legal values depend on the value of the **Type** attribute, as follows:

- For a **datetime** field, the value MUST be **datetime, DateOnly, TimeOnly, ISO8601, ISO8601Basic, ISO8601Gregorian**, or not present. A reader MUST interpret a missing attribute to mean **datetime**.

- For a **Choice** field, the value MUST be **DropDown, RadioButtons, Checkbox**, or not present. A reader MUST interpret a missing attribute to mean **DropDown**.

- For a **CrossProjectLink** field, the value MUST be **EventList** or not present.

- For URL fields, the value MUST be **Image, Hyperlink**, or not present. A reader MUST interpret a missing attribute as **Hyperlink**.

- For a **Boolean** field, the value MUST be **CheckboxIcons, CheckboxIconsWithHeaderIcon**, or not present.

- A reader MUST ignore this attribute for other field types.

**ForcePromoteDemote:** Specifies that the field SHOULD participate in property promotion and demotion.

**FromBaseType:** Indicates that the field was inherited from the list's base type. If TRUE, a server prevents the field from being deleted or having its type changed.
**GridEndNum**: Specifies the largest choice possible in a GridChoice field. A writer MUST include this attribute for GridChoice fields. A reader MUST ignore this attribute for other field types.

**GridNATxt**: Textual value to render for the choice that indicates "not applicable" in a GridChoice field. A reader MUST ignore this attribute for other field types.

**GridStartNum**: Specifies the smallest choice possible in a GridChoice field. A writer MUST include this attribute for GridChoice fields. A reader MUST ignore this attribute for fields of other types. A writer SHOULD set the value of this attribute to 1.

**GridTxtRng1**: Textual value to render for the choice that indicates the value corresponding to that specified by the GridStartNum attribute in a GridChoice field. A reader MUST ignore this attribute for other field types.

**GridTxtRng2**: Textual value to render for the choice that indicates the middle value in a GridChoice field. A reader MUST ignore this attribute for other field types.

**GridTxtRng3**: Textual value to render for the choice that indicates the value corresponding to that specified by the GridEndNum attribute in a GridChoice field. A reader MUST ignore this attribute for fields of other types.

**Group**: Name of the field that is used for grouping purposes. A reader MUST take a localized version of "Custom Columns" as the default value for this attribute.

**HeaderImage**: Specifies a URL, relative to the /_layouts/images folder on the server, to render in place of the field’s DisplayName in views and forms.

**Height**: Height in pixels to render an image for a URL field whose Format attribute is equal to Image. If not present, specifies that the target image does not need to be scaled.

A reader MUST ignore this attribute except for URL fields where the Format attribute specifies Image and the Width attribute is present.

**Hidden**: Specifies whether to render a field in views or forms.

**ID**: GUID for the field.

**Id**: A writer SHOULD NOT include this attribute. A reader MUST ignore this attribute.

**IMEMode**: If specified, indicates a value for the ime-mode attribute to be applied to an <input> tag when reading the field.

**Indexed**: Specifies that a server can optimize for queries that filter on this field.

**ImnHeader**: This attribute is a marker. A reader MUST only check for the existence of the attribute. A writer MUST either not include the attribute or set its value to TRUE.

**IsolateStyles**: For a Text field whose RichText attribute is TRUE and whose RichTextMode attribute is set to FullHtml, this attribute specifies that a server will rewrite the HTML of the field to ensure it will not interfere with the rendering of the surrounding page.

A reader MUST ignore this attribute in other circumstances.

**IsRelationship**: Specifies whether the Lookup field is a relationship lookup field.

A reader MUST ignore this attribute if the field is not a Lookup field.

**JoinColName**: Specifies the physical storage location in the list referred to by the List attribute to compare with the local value of a Lookup field.

**JoinRowOrdinal**: A writer SHOULD NOT include this attribute.<4>
JoinType: Specifies how to treat items that have no corresponding matching item in the list indicated by the List attribute.

A reader MUST ignore this attribute except for Lookup fields.

JumpTo: Specifies the Name of the next field in a survey list when the field has a specified value.

A reader MUST ignore this attribute for Boolean fields.

JumpToFillinChoice: Specifies the Name of the next field in a survey list when a value other than those listed in the CHOICES child element is selected.

A reader MUST ignore this attribute except for Choice fields and MultiChoice fields for which the FillInChoice attribute is TRUE.

JumpToNo: Specifies the Name of the next field in a survey list for a Boolean field for which is false.

JumpToYes: Specifies the Name of the next field in a survey list for a Boolean field for which is true.

LCID: LCID used to render Number fields, Currency fields, and datetime fields. When not present, specifies that the default LCID for the user is to be used. A reader MUST ignore this attribute for other types of fields.

LinkToItem: Specifies whether the field value is rendered as a link to the list item.

LinkToItemAllowed: Specifies the allowed state of the LinkToItem attribute. The allowed values are Prohibited, Required and Allowed. Value Allowed means the link can be optionally shown, Required means the link MUST be shown, Prohibited means the link can't be shown.

List: Specifies the foreign list for a Lookup field.

A writer MUST include this attribute for Lookup fields. A reader MUST ignore this attribute for other kinds of fields.

ListItemMenu: Specifies whether the field value is rendered with a drop-down option menu.

ListItemMenuAllowed: Specifies the allowed state of the ListItemMenu attribute. The allowed values are Prohibited, Required and Allowed. Value Allowed means the menu can be optionally shown, Required means the menu MUST be shown, Prohibited means the menu can't be shown.

Max: Specifies the maximum value for a Number field. A reader MUST ignore this attribute for other kinds of fields.

MaxLength: Specifies the maximum number of characters allowed in a Text field.

MaxLength: The reader MUST ignore this attribute.

Min: Specifies the minimum value for a Number field. A reader MUST ignore this attribute for other kinds of fields.

Mult: In fields whose field internal type is Integer, Number, or Calculated, a reader MUST interpret this as a floating point number by which the value is multiplied at render time. In this context, a reader MUST interpret the default to be 1.0.

In the context of a Lookup field, a reader MUST interpret the presence of this attribute as an indication that the field is a multi-value lookup, and the absence of which as an indication that the field is a single-value lookup. In this context, a writer MUST either omit the attribute or set the value to TRUE.

In other contexts, a reader MUST ignore the attribute and the writer SHOULD NOT include the attribute.
**Name:** String that identifies the field within its list.

**NegativeFormat:** A writer SHOULD NOT include this attribute. A reader MUST ignore this attribute.

**Node:** A writer SHOULD use the **Node** attribute rather than this attribute. A reader MUST ignore this attribute when the **Node** attribute is present. Otherwise, a reader MUST use this attribute for the same purpose as it would use the **Node** attribute.

**Node:** When present, specifies an **XPath** to be used to read or write the value of the field into an **XML document**.

**NoEditFormBreak:** For fields where RichText is TRUE or where the **Type** is Choice or MultiChoice, a reader MUST ignore this attribute. For other fields, if true, this attribute specifies that the following field will be rendered on the same line as this field.

**NumLines:** Number of lines to render when accepting input for a **Note** field. A reader MUST ignore this attribute for other fields.

**Percentage:** When specified on a **Number** field or a **Calculated** field that evaluates to a number, specifies that the field is rendered as a percentage (100 times its value and with a trailing "%" character).

A reader MUST ignore this attribute for other fields.

**PIAttribute:** A reader MUST ignore this attribute if either the **Node** or **node** attribute is present, or if both the **PrimaryPIAttribute** and **PrimaryPITarget** attributes are present, or if the **PITarget** attribute is not present.

Otherwise, the reader MUST treat this attribute as it would the **PrimaryPIAttribute** attribute.

**PIAttribute:** A reader MUST ignore this attribute if either the **Node** or **node** attribute is present, if both the **PrimaryPIAttribute** and **PrimaryPITarget** attributes are present, or if the **PIAttribute** attribute is not present.

Otherwise, the reader MUST treat this attribute as it would the **PrimaryPITarget** attribute.

**PrependId:** Whether the edit form for a **Lookup** field for which the **Multi** attribute is TRUE will present choices sorted by **ID** rather than by **ShowField**.

A reader MUST ignore this attribute in other circumstances.

**Presence:** Whether a **User** field will be decorated with instant messaging presence information.

A reader MUST ignore this attribute for fields that are not user fields.

**PreviousName:** Indicates the value of the **Name** attribute of the field in an earlier revision of the containing list's schema.

**PrimaryKey:** A reader MUST ignore this attribute.

**PrimaryPIAttribute:** When present, specifies an attribute for an XML processing instruction to be used to read or write the value of the field into an XML document.

**PrimaryPITarget:** When present, specifies an XML processing instruction to be used to read or write the value of the field into an XML document.

**ReadOnly:** Whether the user is allowed to change the field through the user interface. If TRUE, only programmatic changes are allowed.
ReadOnlyEnforced: Whether the user is allowed to change the field by any means. If TRUE, the field can only be changed by the system.

RelationshipDeleteBehavior: Describes the relationship, if any, between an item in the list and an item that is the target of a Lookup field.

A reader MUST ignore this attribute when the field is not a Lookup field.

RenderXMLUsingPattern: Whether a Computed field is rendered using its DisplayPattern child element even when rendering for the object model or SOAP interfaces.

A reader MUST ignore this attribute except for Computed fields.

Required: Whether forms presented to accept data for the list item permits blank values for the field.

RestrictedMode: For Notes fields for which the RichText attribute is TRUE, specifies whether the field permits cut, copy, paste, and insert image commands.

For other kinds of fields, the reader MUST ignore this attribute.

ResultType: If the Type attribute specifies that the internal type of the field is Calculated, a reader SHOULD use this attribute as in place of the internal type of the field to render results.

A reader MUST ignore this attribute if the internal type of the field is not Calculated.

ResyncOnChange: Whether the field's value changes value on form submission.

RichText: Whether a Note field contains formatted text; the exact text formatting is subject to the value of the RichTextMode attribute.

A reader MUST ignore this attribute except for Note fields.

RichTextMode: Specifies the serialization of formatted text.

RowOrdinal: The comments for the ColName attribute apply to this attribute as well.

RowOrdinal2: The comments for the ColName attribute apply to this attribute as well.

Sealed: Specifies how the server allows the field to be changed. If TRUE, the server MUST prevent changes to the field except for the DisplayName, Description, Hidden, and Indexed attributes and the ParserRefs child element.

SeparateLine: Determines whether or not the field is displayed on a different row for views that support this feature.

SetAs: A writer SHOULD NOT include this attribute. A reader MUST ignore this attribute.

ShowAddressBookButton: Determines whether or not a field's edit control includes a facility for entering users from an address book. A reader MUST ignore this attribute except when rendering the field in the context of a form that supports this functionality.

ShowField: Specifies the field in the foreign list that supplies the value of a Lookup field. A reader MUST ignore this attribute unless the field is a Lookup field.

If the Type attribute is User, then the default value for this attribute is InmName. If the Type is WorkflowStatus, the default value for this attribute is Status1. Otherwise, the default value is Title.

ShowInDisplayForm: Determines whether or not the field is shown or hidden in a form designed to show the item in a read-only fashion. If TRUE, then the field will be shown in the form for the item. If FALSE, then the field will not be shown in the form. If not specified, an implementation-specific algorithm is used to determine if the field will be shown.
**ShowInEditForm**: When FALSE, indicates that the field is not included in the form that is used to modify an item. When TRUE, indicates that the field's inclusion in such a form depends on implementation.

**ShowInFileDialog**: The reader MUST ignore this attribute.

**ShowInFileDlg**: Similar to **ShowInEditForm** except that it applies to a form designed to collect information about a document from within the context of an application.

**ShowInEditDlg**: Similar to **ShowInEditForm** except that it applies to a form designed to collect information about an item that is being modified.

**ShowInListSettings**: Similar to **ShowInEditForm** except that it applies to the field's inclusion in the user interface that is presented to list schema editors.

**ShowInNewForm**: Similar to **ShowInEditForm** except that it applies to the field's inclusion in a form designed to collect information about an item that is being created.

**ShowInVersionHistory**: Similar to **ShowInEditForm** except that it applies to the field's inclusion in a form designed to display a read-only rendition of a historical version of an item.

**ShowInViewForms**: Similar to **ShowInEditForm** except that it applies to the field's inclusion in the user interface that is presented to view authors.

**Sortable**: When FALSE, specifies that query results are not allowed to be ordered with respect to this field.

**SourceID**: URI suitable for use in an XML namespace in cases where the list schema is transformed to an XSD.

**StaticName**: Local part of an XML element name that is unique within the namespace given by the **SourceID** attribute.

**StorageTZ**: For datetime fields, specifies the time zone in which the field is stored. If TRUE, then UTC is indicated; otherwise, the site's local time zone is indicated.

The reader MUST ignore this attribute unless the field's type is datetime.

**StripWS**: The reader MUST ignore this attribute.

**SuppressNameDisplay**: When specified on a User field, the user's name is not displayed but all other rendering information is.

**TextOnly**: The reader MUST ignore this attribute.

**Title**: When present on a CrossProjectLink fields whose DisplayImage attribute is set and whose Format attribute specifies EventList, this attribute specifies a textual alternative to the image. When not present but the other conditions are specified, the reader MUST infer a default value from the DisplayName attribute. When these conditions are not met, the reader MUST ignore the attribute.

**TitleField**: Specifies that the file is suitable to use for a title of the item.

**Type**: Specifies the rendering properties and internal type of the field.

**UniqueId**: The reader MUST ignore this attribute.

**UnlimitedLengthInDocumentLibrary**: On a Note field or on a Lookup field for which the Mult attribute is TRUE, this specifies that the length of the content is not limited.

**URIComponent**: The reader MUST ignore this attribute.

**URIComponentAsURL**: The reader MUST ignore this attribute.
UserSelectionMode: When present on a User field, PeopleOnly indicates that only principals that are users are selected, and PeopleAndGroups indicates that both principals that are users and those that are groups (2) can be selected.

UserSelectionScope: When present on a User field, specifies the group (2) to which the selected user or users belong.

Version: Current version of the field. The server MUST increment the value by 1 each time the field definition is changed. The client MUST ignore this attribute.

Viewable: Specifies whether or not the field was added to the default view of the list when it was added to the schema of the list. The reader SHOULD ignore this attribute.

WebId: For Lookup fields, specifies the identifier of the site (2) that contains the list referenced by the List attribute.

Width: This attribute has the same semantics as the Height attribute except that it refers to the width of the image rather than the height.

WikiLinking: When TRUE on a Note field whose RichText attribute is TRUE, additional processing is done to facilitate entering hyperlinks within the content of the field. In other circumstances, the reader MUST ignore the attribute.

WorkflowStatusURL: The reader MUST ignore this attribute.

XName: String that is used to correlate the field in an external schema.

2.2.7.3.3.3 Child Elements

Customization: Provides an arbitrary XML document to provide vendor extensibility.

CHOICES: A set of CHOICE string elements that represent a list of available choices for a field. The reader MUST ignore CHOICES if the Type attribute is not Choice or MultiChoice. The writer SHOULD omit CHOICES if the Type attribute is not Choice or MultiChoice.

Default: Default value of instances of data for this field in a list item. The reader MUST ignore this element when the DefaultFormula element is present and not empty.

DefaultFormula: Formula used to calculate the default value for this field in a list item.

DisplayBidiPattern: The implementation-specific XML that specifies the rendering of a Computed field to be used for a site (2) with a locale identifier that specifies a bidirectional read order. A reader MUST ignore DisplayBidiPattern if the Type attribute is not Computed. The writer SHOULD omit DisplayBidiPattern if the Type attribute is not Computed.

DisplayPattern: The implementation-specific XML that specifies the rendering of a Computed field. A reader MUST ignore DisplayPattern if the Type attribute is not Computed. The writer SHOULD omit DisplayPattern if the Type attribute is not Computed.

FieldRefs: Specifies fields needed to render DisplayPattern or DisplayBidiPattern. DisplayPattern and DisplayBidiPattern MUST NOT use fields not listed in this element.

Formula: The formula used to calculate a value for a list item based on the value of other fields in the list. A reader MUST ignore Formula if the Type attribute is not Calculated. A writer SHOULD omit Formula if the Type attribute is not Calculated.

FormulaDisplayNames: The Formula element formatted according to the regional settings of the site (2).

MAPPINGS: A set of MAPPING string elements that represents a canonical, language-agnostic identifier for a corresponding CHOICE with the value specified by the MAPPING element. The reader
MUST ignore MAPPINGS if the Type attribute is not Choice or MultiChoice. The writer SHOULD omit MAPPINGS if the Type attribute is not Choice or MultiChoice.

ParserRefs: Specifies alternate names for the field when accessed by the parsers listed in child elements of this element.

Validation: The implementation-specific XML that specifies the validation criteria to be used for items in this list.

2.2.7.3.4 FieldDefinitionDatabase

The FieldDefinitionDatabase type provides an overall container for field definitions in a Content Database.

2.2.7.3.4.1 Schema

```xml
<xs:complexType name="FieldDefinitionDatabase">
  <xs:sequence>
    <xs:element name="Index" type="IndexDefinitionTP" minOccurs="0" maxOccurs="unbounded" />
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="FieldRef" type="FieldRefDefinitionTP" />
      <xs:element name="Field" type="FieldDefinitionTP" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

2.2.7.3.4.2 Attributes

None.

2.2.7.3.4.3 Child Elements

Index: Specifies a composite field index to be used in this list.

FieldRef: Specifies a reference to an existing field definition that is used in the current list.

Field: Specifies either a definition of a new field definition to be used in this list, or a reference to an existing field that was deleted.

2.2.7.3.5 FieldDefinitionDatabaseWithVersion

The FieldDefinitionDatabaseWithVersion type provides an overall container for field definitions in tp_Fields, specified as follows.

2.2.7.3.5.1 Schema

```xml
<xs:complexType name="FieldDefinitionDatabaseWithVersion" mixed="true">
  <xs:all>
    <xs:element name="tp_Fields" type="FieldDefinitionDatabase" minOccurs="1" maxOccurs="1" />
  </xs:all>
</xs:complexType>
```

2.2.7.3.5.2 Attributes

None.

2.2.7.3.5.3 Child Elements
tp_Fields: Specifies a collection of field definitions that are stored in a content database.

<Content>: The body text of this element MUST contain a string with the following pattern and MUST precede any child elements.

\[
\{[0-9][0-9]*[.]) \{[0-9][0-9]*[.]) \{[0-9] \{[0-9]*[.]) \{[0-9][0-9]*[.]) \{[0-9-9][0-9]*[\)]\}
\]

The first two groups MUST correspond to the major version and minor version of the product. The second two groups MUST correspond to the revision version and build number of the product. The last two groups MUST correspond to the revision of the field definitions on the list and the list's template, respectively.

2.2.7.3.6 FieldDefinitionTP

The FieldDefinitionTP type specifies a field and its associated components. FieldDefinitionTP has the same structure as FieldDefinition. However, if only the ID attribute of FieldDefinitionTP is specified, the element specifies a field definition that was deleted by a user on a front-end Web Server.

2.2.7.3.6.1 Schema

```
<xs:complexType name="FieldDefinitionTP">
  <xs:complexContent>
    <xs:extension base="FieldDefinition">
      <xs:attribute name="Type" type="xs:string" use="optional" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

2.2.7.3.6.2 Attributes

See FieldDefinition (section 2.2.7.3).

2.2.7.3.6.3 Child Elements

See FieldDefinition (section 2.2.7.3).

2.2.7.3.7 FieldParserRef

The FieldParserRef type allows schema authors to override the Name attribute of a FieldDefinition type (section 2.2.7.3.3) for a specific parser.

2.2.7.3.7.1 Schema

```
<xs:complexType name="FieldParserRef">
  <xs:attribute name="Name" type="xs:string" />
  <xs:attribute name="ProgId" type="xs:string" />
</xs:complexType>
```

2.2.7.3.7.2 Attributes

Name: Specifies an alternative Name to provide to the parser specified by the ProgId attribute when describing the field.

ProgId: Specifies the parser to which to provide the alternative Name.
2.2.7.3.7.3 Child Elements

None.

2.2.7.3.8 FieldParserRefs

The FieldParserRefs type allows schema authors to override the Name attribute of a FieldDefinition type (section 2.2.7.3) for a collection of parsers.

2.2.7.3.8.1 Schema

```xml
<xs:complexType name="FieldParserRefs">
  <xs:sequence>
    <xs:element name="ParserRef" type="FieldParserRef" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

2.2.7.3.8.2 Attributes

None.

2.2.7.3.8.3 Child Elements

ParserRef: Specifies the overridden Name for a specific parser.

2.2.7.3.9 FieldRefDefinitionField

The FieldRefDefinitionField type specifies field definitions that are referenced within another field definition.

2.2.7.3.9.1 Schema

```xml
<xs:complexType name="FieldRefDefinitionField" mixed="true">
  <xs:attribute name="Name" type="xs:string" use="required" />
  <xs:attribute name="ID" type="UniqueIdentifierWithOrWithoutBraces" use="optional" />
  <xs:attribute name="ShowField" type="xs:string" use="optional" />
  <xs:attribute name="RefType" type="FieldRefType" use="optional" />
  <xs:attribute name="CreateURL" type="xs:string" use="optional" />
  <xs:attribute name="Key" type="xs:string" use="optional" />
  <xs:attribute name="DisplayName" type="xs:string" use="optional" />
  <xs:attribute name="Format" type="xs:string" use="optional" />
</xs:complexType>
```

2.2.7.3.9.2 Attributes

See attributes section of FieldDefinition (section 2.2.7.3).

Name: Specifies the Name attribute of the referenced field.

ID: Specifies the ID attribute of the referenced fields. When both ID and Name are specified, the reader MUST use ID first and fall back to Name if the ID does not match.

ShowField: Specifies an alternate value for the ShowField attribute on a lookup field when rendering in the context of a computed field.

RefType: It describes the type of reference of the field in an events list. This MUST be a FieldRefType as specified in section 2.2.7.2.4. In other cases, this attribute MUST NOT be present.
**CreateURL:** The URL to create a Meeting Workspace site. If the **RefType** is LinkURL, this attribute MUST be present. Otherwise it MUST NOT be present.

**Key:** If this value of this attribute is set to "Primary", the server MUST give this field priority in the ordering of the items.

**DisplayName:** The reader MUST ignore this attribute.

**Format:** This attribute is only used if the referenced field is DateTime type. If the value of this attribute is set to "Original", the referenced field value will not get rendered.

### 2.2.7.3.9.3 Child Elements

**<content>** Describes the Meeting Workspace site created by the URL in CreateURL. If the **RefType** attribute is present and has the value LinkURL, then the element MUST have content. In other cases, the reader MUST ignore any content.

### 2.2.7.3.10 FieldRefDefinitionIndex

The **FieldRefDefinitionIndex** type specifies a field to use in a composite field index.

#### 2.2.7.3.10.1 Schema

```xml
<xs:complexType name="FieldRefDefinitionIndex">
  <xs:attribute name="ID" type="UniqueIdentifierWithBracesUppercase" use="required" />
</xs:complexType>
```

#### 2.2.7.3.10.2 Attributes

**ID:** Identifies a field for which the server can optimize queries.

#### 2.2.7.3.10.3 Child Elements

None.

### 2.2.7.3.11 FieldRefDefinitionTP

The **FieldRefDefinitionTP** type specifies a reference to a field definition. The attributes specified here override existing values in the field definition.

#### 2.2.7.3.11.1 Schema

```xml
<xs:complexType name="FieldRefDefinitionTP">
  <xs:attribute name="Name" type="xs:string" use="required" />
  <xs:attribute name="ColName" type="xs:string" use="optional" />
  <xs:attribute name="ColName2" type="xs:string" use="optional" />
  <xs:attribute name="RowOrdinal" type="xs:int" default="0" use="optional" />
  <xs:attribute name="RowOrdinal2" type="xs:int" default="0" use="optional" />
  <xs:attribute name="ID" type="UniqueIdentifierWithOrWithoutBraces" use="optional" />
  <xs:attribute name="SourceID" type="xs:string" use="optional" />
  <xs:attribute name="StaticName" type="xs:string" use="optional" />
</xs:complexType>
```

#### 2.2.7.3.11.2 Attributes

See **FieldDefinition** (section 2.2.7.3.3).

#### 2.2.7.3.11.3 Child Elements

See **FieldDefinition** (section 2.2.7.3.3).
None.

2.2.7.3.12  IndexDefinitionTP

The IndexDefinitionTP type specifies a composite field index. A server can optimize for queries that filter on the fields specified within the composite field index.

2.2.7.3.12.1  Schema

```xml
<xs:complexType name="IndexDefinitionTP">
  <xs:sequence>
    <xs:element name="FieldRef" type="FieldRefDefinitionIndex" minOccurs="2" maxOccurs="2" />
  </xs:sequence>
  <xs:attribute name="ID" type="UniqueIdentifierWithBracesUppercase" use="required" />
</xs:complexType>
```

2.2.7.3.12.2  Attributes

ID: Identifier for the composite field index.

2.2.7.3.12.3  Child Elements

FieldRef: Specifies a reference to an existing field definition that is used in the current list.

2.2.7.3.13  MAPPINGDEFINITION

The MAPPINGDEFINITION type defines a canonical value for localizable CHOICE entries. Each MAPPINGDEFINITION MUST define in its contents a corresponding value from a choice.

2.2.7.3.13.1  Schema

```xml
<xs:complexType name="MAPPINGDEFINITION">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="Value" type="xs:string" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

2.2.7.3.13.2  Attributes

Value: String which contains a canonical, non-localizable value for a CHOICE.

2.2.7.3.13.3  Child Elements

<content>: Contains a string that MUST specify exactly the string of a corresponding CHOICE.

2.2.7.3.14  MAPPINGDEFINITIONS

The MAPPINGDEFINITIONS type is a container for one or more MAPPINGS. MAPPINGS MUST NOT be defined for field types other than Choice or MultiChoice. There MUST be either no MAPPINGDEFINITION elements defined for a Choice field, or exactly one MAPPING element for each corresponding CHOICE element.

2.2.7.3.14.1  Schema

```xml
<xs:complexType name="MAPPINGDEFINITIONS">
  <xs:sequence>
  </xs:sequence>
</xs:complexType>
```
2.2.7.3.14.2 Attributes
None.

2.2.7.3.14.3 Child Elements
MAPPING: A canonical value mapping for a CHOICE.

2.2.7.4 Elements
This specification does not define any common XML schema element definitions.

2.2.7.5 Attributes
This specification does not define any common XML schema attribute definitions.

2.2.7.6 Groups
This specification does not define any common XML schema group definitions.

2.2.7.7 Attribute Groups
This specification does not define any common XML schema attribute group definitions.

2.2.8 User-Defined Table Types

2.2.8.1 tvpBSNMetadata
The tvpBSNMetadata table type represents an array of values to identify a stream binary piece. The tvpBSNMetadata table type is defined using T-SQL syntax, as follows.

```t-SQL
TYPE tvpBSNMetadata AS TABLE (  
    DocId uniqueidentifier NULL,  
    Partition tinyint NULL,  
    BSN bigint NULL,  
    StreamId bigint NULL,  
    Type tinyint NULL,  
    Size int NULL  
);  
```

**DocId:** The document identifier for the document that is associated with the stream binary piece.

**Partition:** The stream partition that contains the stream binary piece.

**BSN:** The BLOB sequence number of the stream binary piece.

**StreamId:** The stream identifier of the stream binary piece.

**Type:** The stream type of the stream binary piece.
**Size:** The size, in bytes, of the stream binary piece.

### 2.2.8.2 tvpBSNMetadata2

The `tvpBSNMetadata2` table type represents an array of values to identify a stream binary piece.

The `tvpBSNMetadata2` table type is defined using T-SQL syntax, as follows.

```t-sql
TYPE tvpBSNMetadata2 AS TABLE (  
    DocId uniqueidentifier NULL,  
    Partition tinyint NULL,  
    BSN bigint NULL,  
    StreamId bigint NULL,  
    Type tinyint NULL,  
    Size int NULL,  
    Expiration datetime NULL  
);
```

**DocId:** The document identifier for the document that is associated with the stream binary piece.

**Partition:** The stream partition that contains the stream binary piece.

**BSN:** The BLOB sequence number of the stream binary piece.

**StreamId:** The stream identifier of the stream binary piece.

**Type:** The stream type of the stream binary piece.

**Size:** The size, in bytes, of the stream binary piece.

**Expiration:** The timestamp in UTC format of when the stream binary piece expires.

### 2.2.8.3 tvpStreamData

The `tvpStreamData` Table Type represents an array of values to identify a stream binary piece and its data. It is defined using T-SQL syntax, as follows.

```t-sql
TYPE tvpStreamData AS TABLE (  
    BSN bigint NULL,  
    Data varbinary(max) NULL,  
    Offset bigint NULL,  
    Length int NULL,  
    RbsId varbinary(64) NULL  
);
```

**BSN:** The BLOB sequence number of the stream binary piece.

**Data:** If the stream binary piece is not stored in remote BLOB storage, this value MUST contain a subset of the binary data of the stream binary piece, otherwise it MUST be NULL. See definition for format.

**Offset:** The offset into the stream binary piece where this subset data belongs.

**Length:** The size, in bytes, of this subset data of the stream binary piece.

**RbsId:** If this stream binary piece is stored in remote BLOB storage, this value MUST contain the remote BLOB storage identifier of a subset of the binary data of the stream binary piece. Otherwise it MUST be NULL. For additional information regarding remote BLOB storage, see [MS-WSSO] section 2.1.2.3.8.
3 Protocol Details

3.1 Back-End Database Server Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model, as long as their external behavior is consistent with the behavior described in this document.

The back-end database server maintains the following sets of data for this protocol within both a Configuration Database and one or more content databases. Data within these databases is maintained until updated or removed.

Configuration Objects: A set of information about farm configuration. The Configuration Objects are stored in a Configuration Database.

Site Map: A set of information mapping URLs for site collections to site collection identifiers, and the content databases that contain each site collection's data. The Site Map is stored in the Configuration Database. Site Map entries are identified by site collection identifiers and also are represented by either absolute URLs or store-relative form URLs.

Versions: A set of information indicating the current version information for various components in the farm.

Site Collections: A set of information about all site collections in a content database. Site collection entries are identified by site collection identifiers and are also represented by either absolute URLs or store-relative form URLs.

Sites: A set of information about all sites in a content database. Site entries are identified by site identifiers and are also represented by store-relative form URLs.

Lists: A set of information about all lists (1) in a content database. List entries are identified by list identifiers and are also represented by store-relative form URLs.

List Items: A set of information about all list items in a content database. List item entries are identified by list item identifiers.

Documents: A set of information about all documents in a content database. Document entries are identified by document identifiers and are also represented by store-relative form URLs.

Users: A set of information about all users in a content database. User entries are identified by user identifiers.

Site Groups: A set of information about all permission levels in a content database. Site group entries are identified by site group identifiers.

Roles: A set of information about all roles in a content database. Role entries are identified by role identifiers.

3.1.2 Timers

An execution timeout timer is set up on the back-end database server to govern the execution time for any requests. The amount of time is governed by a time-out value configured on the back-end database server for all connections.
3.1.3 Initialization

Authentication of the TDS connection to the back-end database server MUST occur before this protocol can be used.

The data structures, stored procedures, and actual data are persisted by the back-end database server within databases, so any operations to initialize the state of the database MUST occur before the back-end database server can use this protocol. This protocol requires that the data for site collections, sites, lists, and document libraries already exist within the back-end database server in a valid state.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The T-SQL syntax for each Stored Procedure and Result Set, and the variables they are composed of, is defined using the T-SQL language specified in [TSQL-Ref]. In the T-SQL syntax, the variable name is followed by the type of the variable, which can optionally have a length value in brackets and can optionally have a default value indicated by an equal sign followed by the default value. Unless otherwise specified, all Stored Procedures defined in this section are located in the Content Database.

For definitional clarity, a name has been assigned to any columns in the result sets that do not have a defined name in their current implementation. This does not affect the operation of the result set, because the front-end Web server can access any column with no defined name by ordinal position. Such names are designated in the text using braces in the form {name}. For interoperability, named columns in Result Sets are specified with what they SHOULD be named, and columns marked with braces SHOULD have no defined name. front-end Web server implementations MUST NOT rely on any column name in a Result Set.

The logical sequence of returned values and result sets are indicated in each of the individual Stored Procedures defined in this section. The TDS protocol controls the actual order and structure of how the T-SQL language formatted information is transported over the wire.

All functions, result sets, and stored procedures are defined using T-SQL. The following table lists the functions, result sets, and stored procedures in this section.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fn_GetFullUrl</td>
<td>Constructs a full URL from two component parts.</td>
</tr>
<tr>
<td>proc_AddBuildDependency</td>
<td>Associates a build dependency with a specified document.</td>
</tr>
<tr>
<td>proc_AddDocument</td>
<td>Adds a document to the back-end database server with the specified parameters.</td>
</tr>
<tr>
<td>proc_AddListItem</td>
<td>Adds a list item to a list.</td>
</tr>
<tr>
<td>proc_ChangeLevelForDoc</td>
<td>Updates the publishing level of a document.</td>
</tr>
<tr>
<td>proc_CheckRbsInstalled</td>
<td>Tests whether the back-end database server can support remote BLOB storage.</td>
</tr>
<tr>
<td>proc_CheckoutDocument</td>
<td>Places a short-term lock on a document, refreshes, converts or releases an existing short-term lock, or checks out a document.</td>
</tr>
<tr>
<td>proc_ClearLinks</td>
<td>Deletes all link information associated with a</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>proc_CreateDir</td>
<td>Creates a directory or folder with a specified name at a specified location.</td>
</tr>
<tr>
<td>proc_DeleteAllDocumentVersions</td>
<td>Deletes either the draft versions or older published versions of a document and optionally places them in the recycle bin.</td>
</tr>
<tr>
<td>proc_DeleteDocBuildDependencySet</td>
<td>Deletes a build dependency set for a specified document.</td>
</tr>
<tr>
<td>proc_DeleteDocumentVersion</td>
<td>Deletes a document version and optionally places it in the recycle bin.</td>
</tr>
<tr>
<td>proc_DeleteUrl</td>
<td>Deletes a document.</td>
</tr>
<tr>
<td>proc_DisableRbs</td>
<td>Prepares the back-end database server to stop storing content in remote BLOB storage, and reclaims any back-end database server resources allocated with proc_EnableRbs.</td>
</tr>
<tr>
<td>proc_DirtyDependents</td>
<td>Marks all items that depend on a given document, configuration setting, navigation structure, or usage statistic as &quot;dirty&quot;, so that subsequent action can be taken to update them as necessary.</td>
</tr>
<tr>
<td>proc_EnableRbs</td>
<td>Prepares the back-end database server to store content in remote BLOB storage.</td>
</tr>
<tr>
<td>proc_EnumLists</td>
<td>Returns a list of the lists (1) in a site (2), along with their associated metadata.</td>
</tr>
<tr>
<td>proc_ReadStream</td>
<td>Reads a specific range of data from a document's stream binary piece.</td>
</tr>
<tr>
<td>proc_FetchDocForHttpGet</td>
<td>Fetches a document stream and provides information necessary to render the document on a front-end Web server.</td>
</tr>
<tr>
<td>proc_FetchDocForRead</td>
<td>Requests the metadata information and document stream of a document.</td>
</tr>
<tr>
<td>proc_FetchDocForUpdate</td>
<td>Requests document content and metadata information. It also updates the @CacheParseId flag for the requested document if the @CacheParse parameter is set to 1.</td>
</tr>
<tr>
<td>proc_FetchWelcomeNames</td>
<td>Lists all the names of the default welcome pages used as redirection targets when folders are requested by an HTTP GET in all site collections in the back-end database server.</td>
</tr>
<tr>
<td>proc_GenerateNextId</td>
<td>Returns an identifier to be used for a new list item in a specified list, and to increment the value of the identifier returned by the next call to this procedure for the same List.</td>
</tr>
<tr>
<td>proc_GetAllRolesForUser</td>
<td>Retrieves the role identifiers for the roles assigned to a set of principals in a given security context.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>proc_GetAuditMask</td>
<td>Identifies Audit Flags (section 2.2.2.1) information for a specified object.</td>
</tr>
<tr>
<td>proc_GetAuditMaskOutput</td>
<td>Gets Audit Flags (section 2.2.2.1) information about an object.</td>
</tr>
<tr>
<td>proc_GetContainingList</td>
<td>Gets metadata and event receiver information about the list containing a specified URL.</td>
</tr>
<tr>
<td>proc_GetContainingListCore</td>
<td>Gets metadata and event receiver information about the list containing a specified URL.</td>
</tr>
<tr>
<td>proc_GetDocsMetaInfo</td>
<td>Requests document metadata information for up to ten documents within a specified site (2).</td>
</tr>
<tr>
<td>proc_GetLinkInfoSingleDoc</td>
<td>Provides link information and status for all forward links within a specified document and for all backward links within the specified site collection to the document.</td>
</tr>
<tr>
<td>proc_GetListCheckedOutFiles</td>
<td>Retrieves a list of all documents with a Document Store Type (section 2.2.2.4) of 0 (File) within a specified list, folder, or its subfolders, which are checked out and do not have checked in draft or published versions.</td>
</tr>
<tr>
<td>proc_GetListFields</td>
<td>Gets the mapping of fields in a list.</td>
</tr>
<tr>
<td>proc_GetListMetaDataAndEventReceivers</td>
<td>Retrieves information about the metadata, security scopes, Web Parts, and event receivers for a specified list.</td>
</tr>
<tr>
<td>proc_getObject</td>
<td>Retrieves a single Configuration Object (section 2.2.5.1) from the Configuration Database (section 3.1.1).</td>
</tr>
<tr>
<td>proc_getObjectsByBaseClass</td>
<td>Returns a list of GUIDs for child Configuration Objects (section 2.2.5.1) of the specified parent Configuration Object that inherit from the specified base Class (section 2.2.5.1.1).</td>
</tr>
<tr>
<td>proc_getObjectsByClass</td>
<td>Retrieves the GUIDs of Configuration Objects (section 2.2.5.1) based upon the Class type (section 2.2.5.1.1), the parent Configuration Object, and the Name (section 2.2.5.1.3) of the Configuration Object.</td>
</tr>
<tr>
<td>proc_GetSiteFlags</td>
<td>Returns the Site Collection Flags (section 2.2.2.9) set for a specified site collection.</td>
</tr>
<tr>
<td>proc_getSiteMap</td>
<td>Determines the Site Map (section 3.1.5.38.1) information for a site collection.</td>
</tr>
<tr>
<td>proc_getSiteMapById</td>
<td>Determines the complete Site Map (section 3.1.5.38.1) information for a site collection.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>proc_GetTpWebMetaDataAndListMetaData</td>
<td>3.1.5.38.1) information for a site collection.</td>
</tr>
<tr>
<td>proc_GetUniqueScopesInList</td>
<td>Retrieves metadata for a particular site (2) or list (1).</td>
</tr>
<tr>
<td>proc_GetVersion</td>
<td>Gets the component version number string associated with the specified version identifier GUID.</td>
</tr>
<tr>
<td>proc_GetWebMetaInfo</td>
<td>Requests metadata information for a site (2).</td>
</tr>
<tr>
<td>proc_GetWebMetainfoByUrl</td>
<td>Requests site metadata information for the site (2) containing a specified Uniform Resource Locator (URL).</td>
</tr>
<tr>
<td>proc_ListDocumentVersions</td>
<td>Lists all available version history information for a specified document.</td>
</tr>
<tr>
<td>proc_ListRbsStores</td>
<td>Enumerates the remote BLOB storage stores with which the back-end database server has been configured, if any.</td>
</tr>
<tr>
<td>proc_ListUrls</td>
<td>Retrieves metadata for a document specified by a URL and the documents contained within it, if any.</td>
</tr>
<tr>
<td>proc_RenameUrl</td>
<td>Renames a document or folder within a specified site (2).</td>
</tr>
<tr>
<td>proc_SecAddPrincipalToRole</td>
<td>Adds a security principal to a role defined within a site collection.</td>
</tr>
<tr>
<td>proc_SecAddRoleDef</td>
<td>Creates a new role definition for a specified site (2) within a site collection.</td>
</tr>
<tr>
<td>proc_SecAddUser</td>
<td>Adds a new entry for a security principal to the UserInfo Table (section 2.2.6.10) that contains descriptive properties and security information about security principals (2), and is stored in the back-end database server.</td>
</tr>
<tr>
<td>proc_SecAddUserToSiteGroup</td>
<td>Adds a user to a permission level in the site collection.</td>
</tr>
<tr>
<td>proc_SecAddWebMembership</td>
<td>Associates an existing user within a site collection to a site (2) within the same site collection.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>proc_SecChangeToInheritedList</strong></td>
<td>Changes the scope of the specified list to the specified site (2) and remove any role assignments and permission settings specific to the list.</td>
</tr>
<tr>
<td><strong>proc_SecChangeToInheritedWeb</strong></td>
<td>Changes a site (2) from having its own unique permissions to instead use the permissions inherited from its nearest ancestor with unique permissions.</td>
</tr>
<tr>
<td><strong>proc_SecChangeToUniqueScope</strong></td>
<td>Sets a securable object such as a site (2), list, or document library to use its own unique security scope, instead of inheriting its security scope from the first ancestor with uniquely permissions.</td>
</tr>
<tr>
<td><strong>proc_SecCheckDeletedAccounts</strong></td>
<td>Checks whether a login name exists in the site collection.</td>
</tr>
<tr>
<td><strong>proc_SecCloneRoleDefinitions</strong></td>
<td>Creates a copy of the current role definition for a site (2).</td>
</tr>
<tr>
<td><strong>proc_SecCreateSiteGroup</strong></td>
<td>Adds a new permission level to a site collection.</td>
</tr>
<tr>
<td><strong>proc_SecDecCurrentUsersCount</strong></td>
<td>Reduces by one the total number of users in the specified site collection when configured to use Active Directory account creation mode.</td>
</tr>
<tr>
<td><strong>proc_SecGetAccountStatus</strong></td>
<td>Provides status information for a site collection's users, which are not marked as deleted, and match the specified login name or email address.</td>
</tr>
<tr>
<td><strong>proc_SecGetAclFromScope</strong></td>
<td>Obtains the Access Control List and the anonymous User permissions for a given scope.</td>
</tr>
<tr>
<td><strong>proc_SecGetAllAclsForSite</strong></td>
<td>Lists all Scope Identifiers and their associated ACL and anonymous permission masks in a site collection.</td>
</tr>
<tr>
<td><strong>proc_SecGetAllGroupsAndMembershipInfo</strong></td>
<td>Returns information about all site groups and site group members within a site collection.</td>
</tr>
<tr>
<td><strong>proc_SecGetApplicationPrincipalAndUserToken</strong></td>
<td>Returns information about a principal (1) based on the principal's login name and user identifier.</td>
</tr>
<tr>
<td><strong>proc_SecGetCompleteWebRoleMemberList</strong></td>
<td>Lists all role assignments for all the principals with permissions on a specified Site.</td>
</tr>
<tr>
<td><strong>proc_SecGetCurrentUsersCount</strong></td>
<td>Returns a result set containing a count of Users in the specified site collection.</td>
</tr>
<tr>
<td><strong>proc_SecGetDomainGroupMapData</strong></td>
<td>Retrieves the domain group map cache.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>proc_SecGetGroupById</td>
<td>Checks whether the specified Group (either a site group or a domain group) exists in the specified site collection.</td>
</tr>
<tr>
<td>proc_SecGetGroupOwner</td>
<td>Retrieves the User Identifier or Site Group Identifier for the owner of a site group.</td>
</tr>
<tr>
<td>proc_SecGetGroupSecurityScopes</td>
<td>Retrieves a site group’s role assignments information on all security scopes within a given site collection.</td>
</tr>
<tr>
<td>proc_SecGetIndividualUrlSecurityCheckEventReceivers</td>
<td>Requests security information and event receiver information about a document at a specified location.</td>
</tr>
<tr>
<td>proc_SecGetItemsWithUniquePermissions</td>
<td>Returns information about list items with unique permissions.</td>
</tr>
<tr>
<td>proc_SecGetPrincipalByEmail</td>
<td>Returns user information about a user associated with a specified email address.</td>
</tr>
<tr>
<td>proc_SecGetPrincipalById</td>
<td>Returns information about a principal or collection of principals based on a specified site group identifier or user identifier.</td>
</tr>
<tr>
<td>proc_SecGetPrincipalByIdEx</td>
<td>Returns information about a principal based on a specified user identifier or a collection of principals based on a specified site group identifier.</td>
</tr>
<tr>
<td>proc_SecGetPrincipalByLogin</td>
<td>Returns security and attribute information for a principal (a user or domain group) identified by a specified login name.</td>
</tr>
<tr>
<td>proc_SecGetPrincipalByLogin20</td>
<td>Returns security principal (2) information based on up to 20 separate login names.</td>
</tr>
<tr>
<td>proc_SecGetPrincipalDisplayInformation20</td>
<td>Returns security principal (2) or site group information for up to 20 principal identifiers.</td>
</tr>
<tr>
<td>proc_SecGetRoleAssignments</td>
<td>Requests the WSS ACE (section 2.2.3.5) for a specified security scope in a site collection.</td>
</tr>
<tr>
<td>proc_SecGetRoleBindingsForAllPrincipals</td>
<td>Lists the role assignments for all principals in a security scope.</td>
</tr>
<tr>
<td>proc_SecGetRoleDefs</td>
<td>Retrieves role definition information for items in the specified site collection and scope.</td>
</tr>
<tr>
<td>proc_SecGetSecurityInfo</td>
<td>Retrieves security permissions information about</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>proc_SecGetSiteAdmins</td>
<td>Returns a list of all site collection administrators not marked as deleted.</td>
</tr>
<tr>
<td>proc_SecGetSiteGroupById</td>
<td>Gets information about a site group given its ID.</td>
</tr>
<tr>
<td>proc_SecGetSiteGroupByTitle</td>
<td>Gets site group information for the site group with the specified user-friendly name.</td>
</tr>
<tr>
<td>proc_SecGetSiteGroupByTitle20</td>
<td>Provides information about site groups in bulk, as specified by a set of up to 20 site group titles.</td>
</tr>
<tr>
<td>proc_SecGetUserAccountDirectoryPath</td>
<td>Returns the User Account Directory Path of a specified site collection.</td>
</tr>
<tr>
<td>proc_SecGetUserPermissionOnGroup</td>
<td>Determines what permissions the requesting user has within a specified site group.</td>
</tr>
<tr>
<td>proc_SecGetWebsAndListWithUniquePermissions</td>
<td>Returns a list of sites and lists that have unique permissions.</td>
</tr>
<tr>
<td>proc_SecListAllSiteMembers</td>
<td>Provides information about all non-deleted security principals (2) in the specified site collection.</td>
</tr>
<tr>
<td>proc_SecListAllWebMembers</td>
<td>Lists all non-deleted user and domain group accounts registered with a specified site (2).</td>
</tr>
<tr>
<td>proc_SecListGroupsInRole</td>
<td>Lists all site groups that have the specified role.</td>
</tr>
<tr>
<td>proc_SecListScopeGroups</td>
<td>Lists all site groups that have role assignments in a specified scope.</td>
</tr>
<tr>
<td>proc_SecListScopeUsers</td>
<td>Lists information about users and domain groups with a direct role assignment association with the specified scope, rather than by membership in a site group with a role assignment association with the scope.</td>
</tr>
<tr>
<td>proc_SecListSiteGroupMembership</td>
<td>Lists members in a specified site group.</td>
</tr>
<tr>
<td>proc_SecListSiteGroups</td>
<td>Lists site group information for a specified site collection.</td>
</tr>
<tr>
<td>proc_SecListSiteGroupsContainingUser</td>
<td>Lists the site groups in which the user is a member.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>proc_SecListSiteGroupsWhichUserOwns</td>
<td>Returns site group information for the site groups owned by the specified principal.</td>
</tr>
<tr>
<td>proc_SecListUsersInRole</td>
<td>Lists the non-deleted principals assigned to a specified role in the specified scope.</td>
</tr>
<tr>
<td>proc_SecMigrateUser</td>
<td>Updates the SystemID and login name for a principal in the UserInfo Table (section 2.2.6.10) and AllUserData Table (section 2.2.6.2).</td>
</tr>
<tr>
<td>proc_SecReCalculateWebFGP</td>
<td>Updates the bit at 0x00000400 of the Site Property Flags (section 2.2.2.11) to indicate whether the site (2) has at least one uniquely secured object within it.</td>
</tr>
<tr>
<td>proc_SecRefreshToken</td>
<td>Updates the UserInfo Table (section 2.2.6.10) with information about a specified user’s membership in external groups.</td>
</tr>
<tr>
<td>proc_SecRemoveGroup</td>
<td>Removes a site group from a site collection.</td>
</tr>
<tr>
<td>proc_SecMarkGroupForWebDelete</td>
<td>Marks a site group for later removal from a site collection.</td>
</tr>
<tr>
<td>proc_SecRemovePrincipalFromScope</td>
<td>Removes a principal from a security role associated with a site in a specified security scope.</td>
</tr>
<tr>
<td>proc_SecRemoveRoleDef</td>
<td>Removes a role definition, and any role assignments which use that role definition, from a specified site (2).</td>
</tr>
<tr>
<td>proc_SecRemoveUserFromScopeByLogin</td>
<td>Removes a user, specified by login name, from a role in a specified security scope.</td>
</tr>
<tr>
<td>proc_SecRemoveUserFromSite</td>
<td>Removes a user from a site collection.</td>
</tr>
<tr>
<td>proc_SecRemoveUserFromSiteGroup</td>
<td>Removes a user from a site group in a site collection.</td>
</tr>
<tr>
<td>proc_SecRemoveUserFromSiteGroupByLogin</td>
<td>Removes a user identified by login name from a specified site group.</td>
</tr>
<tr>
<td>proc_SecResetItemPerm</td>
<td>Causes a list item to revert its set of permissions so that it inherits permissions from its parent rather than has its own.</td>
</tr>
<tr>
<td>proc_SecResetWebToDefaultRoleDefinition</td>
<td>Replaces existing roles, role assignments, and Permissions for a specified site (2) and its subsites with a default set of role definitions, as specified by input parameters.</td>
</tr>
<tr>
<td>proc_SecResolvePrincipal</td>
<td>Retrieves information about a principal or site group in the specified site collection, given the</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>proc_SecSetSiteGroupProperties</td>
<td>Updates properties of a site group.</td>
</tr>
<tr>
<td>proc_SecSetUserAccountDirectoryPath</td>
<td>Sets or clears the user account directory path for a specified site collection.</td>
</tr>
<tr>
<td>proc_SecSetWebRequestAccess</td>
<td>Sets the request access email address for a specified site (2).</td>
</tr>
<tr>
<td>proc_SecUpdateAnonymousPermMask</td>
<td>Modifies the permissions for the anonymous user.</td>
</tr>
<tr>
<td>proc_SecUpdateDomainGroupMapData</td>
<td>Updates the domain group map cache of a specified site collection.</td>
</tr>
<tr>
<td>proc_SecUpdateRoleDef</td>
<td>Updates a role definition with a new title, description, display order, Role Definition Type (section 2.2.1.2.16), and WSS Rights Mask (section 2.2.2.15).</td>
</tr>
<tr>
<td>proc_SecUpdateUser</td>
<td>Updates the display name, e-mail address, notes, or administrative privileges listed in the user information associated with a principal.</td>
</tr>
<tr>
<td>proc_SecUpdateUserActiveStatus</td>
<td>Marks a user as an active user in a site collection.</td>
</tr>
<tr>
<td>proc_TakeOverCheckOut</td>
<td>Overrides checkout for a document that is checked out and has no checked-in draft or published version.</td>
</tr>
<tr>
<td>proc_UncheckoutDocument</td>
<td>Releases a checked out document.</td>
</tr>
<tr>
<td>proc_UpdateDocBuildDependencySet</td>
<td>Stores an updated version of a build dependency set for a specified document.</td>
</tr>
<tr>
<td>proc_UpdateDocument</td>
<td>Updates the metadata and stream of a document.</td>
</tr>
<tr>
<td>proc_UpdateListItem</td>
<td>Updates a list item in a document library or list.</td>
</tr>
<tr>
<td>proc_UpdateListSettings</td>
<td>Updates list (1) metadata.</td>
</tr>
<tr>
<td>proc_UpdateUserInfoInTableFromRowUpdater</td>
<td>Updates the user information data for the specified user.</td>
</tr>
<tr>
<td>proc_UrlToWebUrl</td>
<td>Returns the store-relative form URL of the site (2) that contains a specified store-relative form URL and is inside a specific site collection.</td>
</tr>
<tr>
<td>proc_WriteChunkToAllDocStreams</td>
<td>Establishes new document content associated with the document specified by @DocId or appends to already existing document content</td>
</tr>
</tbody>
</table>
### Name | Description
--- | ---
| proc_WriteStreamToRBS | Establishes the remote BLOB storage identifier representing a new stream binary piece for a document.
| proc_WriteStreams | Creates a new stream binary piece for a document.
| proc_WriteStreamToSQL | Creates a new stream binary piece for a document.
| proc_SetStreamsToDoc | Associates stream binary pieces with a document.
| proc_SetStreamsToDocNoTVP | Associates stream binary pieces with a document.
| TVF_Docs.Url_Level | Returns a subset of rows from the Docs View (section 2.2.6.3).
| TVF_UserData.ListItemLevelRow | Returns a subset of rows from the UserData View (section 2.2.6.8).
| TVF_UserData.PID_DID_Level_Row | Returns a subset of rows from the UserData View (section 2.2.6.8).
| proc_HasCurrentPublishVersion | Tests whether the specified list item has a current published version.
| proc_GetSiteDenyPermMask | Returns a WSS Rights Mask (section 2.2.15) specifying the rights to deny the current user in the specified site collection.
| proc_GetNewListItemId | Gets the next available row identifier of a list (1) and the number of rows that will be copied or moved when copying or moving a URL to that list (1).
| proc_UpdateDiskUsed | Updates the disk-used size, in Bytes, of a site collection.

### 3.1.5.1 fn_GetFullUrl

The `fn_GetFullUrl` function is invoked to construct a full URL from two component parts.

```sql
FUNCTION fn_GetFullUrl(
    @DirName nvarchar(256),
    @LeafName nvarchar(128))
RETURNS nvarchar(260)
```

- **@DirName**: The directory name component of the full URL.
@LeafName: The leaf name component of the full URL.

Return Values: The full URL, which is formed from @DirName and @LeafName as follows:

If @DirName is an empty string, then @LeafName MUST be returned. If @LeafName is an empty string, then @DirName MUST be returned. If either @DirName or @LeafName is NULL, then NULL MUST be returned. Otherwise fn_GetFullUrl MUST return @DirName + '/' + @LeafName.

3.1.5.2 proc_AddBuildDependency

The proc_AddBuildDependency stored procedure is invoked to associate a build dependency with a specified document.

PROCEDURE proc_AddBuildDependency(
    @DocSiteId                     uniqueidentifier,
    @DocDirName                    nvarchar(256),
    @DocLeafName                   nvarchar(128),
    @TargetDirName                 nvarchar(256),
    @TargetLeafName                nvarchar(128),
    @DirectDependency              bit,
    @RequestGuid                   uniqueidentifier  = NULL OUTPUT
);

@DocSiteId: The Site Collection Identifier (section 2.2.1.1.9) for the site collection that contains the document. This parameter MUST NOT be NULL.

@DocDirName: The directory name of the document in store-relative form. This parameter MUST NOT be NULL.

@DocLeafName: The leaf name of the document. This parameter MUST NOT be NULL.

@TargetDirName: The directory name of the item to declare as a dependency of the document. This parameter MUST NOT be NULL.

@TargetLeafName: The leaf name of the item to declare as a dependency of the document. This parameter MUST NOT be NULL.

@DirectDependency: Set to 1 if the target is a direct dependency. Set to 0 if indirect, such as a dependency of a dependency. This parameter MUST NOT be NULL.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_AddBuildDependency stored procedure MUST return an integer return code of 0. The proc_AddBuildDependency stored procedure MUST NOT return a result set.

3.1.5.3 proc_AddDocument

The proc_AddDocument stored procedure is invoked to add a document to the back-end database server with the specified parameters.

PROCEDURE proc_AddDocument(
    @DocSiteId                     uniqueidentifier,
    @DocWebId                      uniqueidentifier,
    @UserId                        int,
    @AppPrincipalId                int,
    @AuthorId                      int,
    @DocDirName                    nvarchar(256),
    @DocLeafName                   nvarchar(128)     OUTPUT,
    @Level                         tinyint,
    @DocDirName                    nvarchar(256),
    @DocLeafName                   nvarchar(128),
    @DirectDependency              bit,
    @RequestGuid                   uniqueidentifier  = NULL OUTPUT
);
@UIVersion                     int               = 512,
@NewDocId                      uniqueidentifier,
@DocLibId                      uniqueidentifier,
@NewDocLibRowId                int,
@SendingContent                bit,
@DocMetaInfo                   varbinary(max),
@DocSize                       int,
@DocMetaInfoSize               int,
@DocFileFormatMetaInfo         varbinary(max),
@DocFileFormatMetaInfoSize     int,
@EnableMinorVersions           bit,
@IsModerated                   bit,
@DocDirty                      bit,
@DocFlags                      int,
@DocIncomingCreatedDTM          datetime,
@DocIncomingDTM                datetime,
@GetWebListForNormalization    bit,
@PutFlags                      bigint,
@CreateParentDir               bit,
@UrlIsSuggestion               bit,
@ThicketMainFile               bit,
@CharSet                       int,
@ProgId                        nvarchar(255),
@AttachmentOp                  int,
@VirusVendorID                 int,
@VirusStatus                   int,
@VirusInfo                     nvarchar(255),
@VirusInfoEx                   varbinary(max),
@LockTimeout                   int,
@SharedLock                    bit,
@Comment                       nvarchar(1023),
@DocDTM                        datetime          OUTPUT,
@NoQuotaOrLockCheck           bit,
@DocContentVerBump             int,
@BSNBump                       bigint,
@StreamSchema                  tinyint,
@DocClientId                   varbinary(16)     = NULL,
@RequestGuid                   uniqueidentifier  = NULL OUTPUT
)

@DocSiteId: The site collection identifier for the site collection that will contain the document to be stored. This MUST NOT be NULL.

@DocWebId: The site identifier for the site (2) that will contain the document to be stored. This MUST NOT be NULL.

@UserId: The user identifier of the current user making the request to the front-end Web server. This value MUST refer to an existing user identifier for the specified site collection.

@AppPrincipalId: The app principal identifier of the app that is requesting this operation. If the operation was not requested by an app then it MUST be set to 0.

@AuthorId: The user identifier to use instead of @UserId's value in the owner fields of the document, if the publishing level is set to draft or checked out or the document has a short-term lock applied. If NULL is specified, @UserId is used in the owner fields.

@DocDirName: The directory name of the document to be stored. This MUST NOT be NULL.

@DocLeafName: The leaf name of the document to be stored. If @UrlIsSuggestion is set to "1", this name MUST be replaced with a unique name if the name is not unique and is returned in this output parameter as the actual document leaf name. This parameter MUST NOT be NULL.

@Level: The publishing level type value (section 2.2.2.6) for the document to be stored. This MUST be a valid value.
@UIVersion: The UI version number to associate with this document. This MUST NOT be NULL.

@NewDocId: The document identifier (section 2.2.1.1.2) of the document to be stored. This MUST NOT be NULL, MUST be unique for a new document, and MUST be the same for an existing document adding a new publishing level.

@DoclibId: The list identifier (section 2.2.1.1.5) of the list (1) or document library into which the document is to be stored. This MUST only be NULL when a document is not being added to a list (1) or document library.

@NewDoclibRowId: The identifier for a row in a document library for the document to be stored. This MUST be NULL if the document is not in a list (1) or document library.

@SendingContent: Specifies whether to store the document stream in the back-end database server. This field MUST be one if the document stream of the document is intended to be stored in the back-end database server; otherwise, it MUST be zero.

@DocMetaInfo: The metadata information for the document to be stored. If there is no metadata information for this document, this parameter MUST be NULL.

@DocSize: Final size in bytes of the document stream of the document. This MUST be zero if @SendingContent is zero.

@DocMetaInfoSize: Size in bytes of the document's metadata info. This MUST be zero if @DocMetaInfo is NULL.

@DocFileFormatMetaInfo: The DocFileFormatMetaInfo data for the document to be stored. If there is no DocFileFormatMetaInfo data for this document, this parameter MUST be NULL.

@DocFileFormatMetaInfoSize: Size in bytes of the DocFileFormatMetaInfo data. This MUST be zero if @DocFileFormatMetaInfo is NULL.

@EnableMinorVersions: Specifies whether minor versions are enabled on the document. If this parameter is set to "1", minor versions MUST be enabled; otherwise, minor versions MUST NOT be enabled.

@IsModerated: Specifies whether the list (1) or document library into which the document will be stored has content approval enabled. The value MUST be one if the list (1) or document library into which the document will be stored has content approval enabled; otherwise, it MUST be zero. If the document is not to be stored in a list (1) or document library, the value MUST be zero.

@DocDirty: Specifies whether the document has dependencies such as links to other items. If this parameter is set to "1", the document has dependencies that MUST subsequently be updated.

@DocFlags: A Document Flags value (section 2.2.3) that contains options for adding the document.

@DocIncomingCreatedDTM: A time stamp in UTC format that specifies when the document was created.

@DocIncomingDTM: A time stamp in UTC format that specifies the document's last modification date.

@GetWebListForNormalization: Specifies whether to return the Site List for Normalization Result Set (section 3.1.5.3.1). If this parameter is set to "1", this procedure MUST return a list (1) of the immediate child subsites of the document's containing site in the Site List for Normalization Result Set. If this parameter is set to "0", the Site List for Normalization Result Set MUST NOT be returned. This parameter MUST NOT be NULL.

@PutFlags: A Put Flags Type value (section 2.2.7) that specifies the options for adding the document.
@CreateParentDir: Specifies whether to create the parent directory for the document to be added if it does not already exist. If this parameter is set to "1", the parent directory specified by @DocDirName MUST be created if it does not exist. If this parameter is set to "0", this procedure MUST fail if the parent directory does not exist. This parameter MUST NOT be NULL.

@UrlIsSuggestion: Specifies whether the @DocLeafName provided MUST be changed to a unique value if it is not unique. If this parameter is set to "1", @DocLeafName MUST be updated to get a guaranteed unique URL. If this parameter is set to "0" and the URL is not unique, this procedure MUST fail. This parameter MUST NOT be NULL.

@ThicketMainFile: Specifies whether the document is a thicket main file. If this parameter is set to "1", the document is a thicket main file. If this parameter is set to "0", the document is not a thicket main file or a thicket supporting file. If this parameter is NULL, this document is a thicket supporting file.

@CharSet: An optional parameter that specifies a code page for the character set to be associated with the document.

@ProgId: An optional parameter that specifies a preferred application to open the document. The @ProgId value is used to distinguish between different applications that save files with a given file extension (for example, different editors for HTML or XML files).

@AttachmentOp: An attachments flag value (section 2.2.1.2.1) that specifies the security checks to be performed by proc_AddDocument on this document's URL, based on whether it appears to be an attachment.

@VirusVendorId: An optional parameter that specifies the identifier of the virus scanner that processed this document.

@VirusStatus: A virus status type (section 2.2.1.2.17) that specifies the current virus check status of this document.

@VirusInfo: A string containing a provider-specific message returned by the virus scanner when it last processed the document. This value MUST be NULL if the document does not exist or if the document has not been processed by a virus scanner.

@VirusInfoEx: A binary value that specifies extensible virus information for this document, which is opaque to the back-end server.

@LockTimeout: An integer value that specifies the number of minutes remaining to set on a short-term lock on the document. This value MUST be a positive value if a short-term lock is requested, and this procedure MUST set the short-term lock for this period. Otherwise, this procedure MUST NOT set a short-term lock on the document.

@SharedLock: Specifies whether to establish a shared lock on the document after it is added. This parameter MUST be set to one to establish a shared lock; otherwise, it MUST be set to zero.

@Comment: An optional text check-in comment to associate with the document. This value MUST be ignored when @Level is set to "255".

@DocDTM: An output parameter for the time stamp of the last modification date of the document. This parameter MUST be set to the value of @DocIncomingDTM or to the current UTC datetime if @DocIncomingDTM is NULL.

@ifNoQuotaOrLockCheck: Specifies whether to bypass the disk quota and disk lock check. If this parameter is set to "1", the checks are bypassed. If this parameter is set to "0", an explicit check will be made to see if the site (2) is locked or if quota is reached.

@DocContentVerBump: An integer value used to increase the version of the content stored in the back-end database server. This value is added to the current content version number.
@BSNBump: Specifies the amount the BLOB sequence number of the document is increased by.

@StreamSchema: Specifies the stream schema of the document to be created.

@DocClientId: An optional parameter that specifies the client identifier of the document to be created.

@RequestGuid: The optional request identifier for the current request.

Return Values: This procedure returns an integer return code, which MUST be one of the values listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The path specified for the document was not found.</td>
</tr>
<tr>
<td>5</td>
<td>Access denied error while attempting to create a directory.</td>
</tr>
<tr>
<td>80</td>
<td>Attempted to create a directory when the specified directory exists. The document already exists. The attempt to add the document failed. SendingContent was specified, but the document is not checked out. The attempt to add the document stream failed.</td>
</tr>
<tr>
<td>212</td>
<td>The disk storage was locked.</td>
</tr>
<tr>
<td>1816</td>
<td>Disk quota exceeded.</td>
</tr>
</tbody>
</table>

The proc_AddDocument stored procedure returns zero to two result sets in the following order: Site List for Normalization Result Set (section 3.1.5.3.1), Checkout Information Result Set (section 3.1.5.3.2).

3.1.5.3.1 Site List for Normalization Result Set

The Site List for Normalization Result Set returns a list of URLs for the immediate child subsites of the site (2) containing the newly added document. The Site List for Normalization Result Set MUST be produced when the input parameter @GetWebListForNormalization is set to "1" and execution has been successful up to the point of inserting the document. The Site List for Normalization Result Set MUST contain one row for each subsite found. The Site List for Normalization Result Set is defined in the Common Result Sets URL Result Set (section 2.2.4.27).

3.1.5.3.2 Checkout Information Result Set

The Checkout Information Result Set returns check-out information about the document. The Checkout Information Result Set MUST be returned on successful completion if either the input parameter @Level is set to "255", indicating that the document is checked out, or if the input parameter @LockTimeout is not NULL, indicating that a short-term lock was applied. The Checkout Information Result Set MUST contain one row.

```sql
    tp_Login nvarchar(255),
    CheckoutDate datetime,
    {CheckoutExpires} datetime;
```

tp_Login: The login name of the principal (1) to whom the document is checked out. If the document is currently checked in, this MUST be NULL.
CheckoutDate: A time stamp, in UTC format, indicating when this document was checked out. If the document is currently checked in, this MUST be NULL.

{CheckoutExpires}: A time stamp, in UTC format, indicating when the short-term lock for this document will expire. If the document is currently checked in or has a long-term checkout, this MUST be NULL.

3.1.5.4 proc_AddListItem

The proc_AddListItem stored procedure is invoked to add a list item to a list (1).

PROCEDURE proc_AddListItem (  
@SiteId                        uniqueidentifier,  
@WebId                         uniqueidentifier,  
@ListID                        uniqueidentifier,  
@RowOrdinal                    tinyint,  
@ItemId                        int                           OUTPUT,  
@UserId                        int,  
@Size                          int,  
@TimeNow                       datetime,  
@CopySecurityFromMasterID      int                = NULL,  
@ExtraItemSize                 int                = NULL,  
@CheckDiskQuota                bit                = 1,  
@ItemDocType                   tinyint            = 0,  
@SortTypeReversed              bit                = 0,  
@BaseRowItemId                 int                = NULL,  
@DocIdAdded                    uniqueidentifier   = NULL,  
@RetainId                      uniqueidentifier   = NULL,  
@RetainObjectIdentity          bit                = 0,  
@Level                         tinyint            = 1,  
@UIVersion                     int                = 512,  
@ItemCountDelta                int                = 1,  
@ItemName                      nvarchar(255)      = NULL,  
@UseNvarchar1ItemName          bit                = 1,  
@ItemDirName                   nvarchar(256)      = NULL,  
@ServerTemplate                int                = NULL,  
@IsNotUserDisplayed            bit                = NULL,  
@BaseType                      int                = NULL,  
@CheckSchemaVersion            int                = NULL,  
@OnRestore                     bit                = NULL,  
@AddNamespace                  bit                = 0,  
@tp_Ordering                   varchar(512)       = NULL,  
@tp_ThreadIndex                 varbinary(512)     = NULL,  
@tp_HasAttachment              bit                = NULL,  
@tp_ModerationStatus           int                = 0,  
@tp_IsCurrent                   bit                = 1,  
@tp_ItemOrder                   float              = NULL,  
@tp_InstanceID                  uniqueidentifier   = NULL,  
@tp_GUID                       uniqueidentifier   = NULL,  
@tp_Id                         int                = NULL,  
@tp_Author                     int                = NULL,  
@tp_Editor                     int                = NULL,  
@tp_Created                    datetime           = NULL,  
@tp_Modified                   datetime           = NULL,  
@tp_Created                    datetime           = NULL,  
@tp_Version                    int                = 1,  
@tp_ContentTypeId              varbinary(512)       = NULL,  
@tp_CopySource                 nvarchar(260)      = NULL,  
@tp_HasCopyDestinations        bit                = NULL,  
@tp_WorkflowVersion            int                = 1,  
@tp_WorkflowInstanceId         uniqueidentifier   = NULL,  
@nvarchar1                      nvarchar(255)      = NULL,  
@MtgEventType                  int                = NULL,  
@EventUID                      nvarchar(255)      = NULL,  
@RecurrenceID                  datetime           = NULL,
@SiteId: The site collection identifier for the site collection containing the list (1) that the list item is being added to.

@WebId: The site identifier for the site containing the list (1) that the list item is being added to.

@ListID: The list identifier of the list (1) that the list item is being added to.

@RowOrdinal: The 0-based ordinal index of the current row to add for this list item in the set of rows representing the list item in the AllUserData table (section 2.2.6.2). If a list item requires multiple rows to represent it in the AllUserData table because it contains more defined data columns than will fit in a single row, the front-end Web server MUST call proc_AddListItem again, with the additional data using the next row value in the @RowOrdinal parameter. This parameter MUST NOT be NULL.

@ItemId: An output parameter that returns the identifier of the list item that has been added:

- If @tp_ID is not NULL and @BaseRowItemID is NULL, proc_AddListItem MUST use the value specified by @tp_Id.
- If @BaseRowItemID is not NULL, proc_AddListItem MUST use the value specified by @BaseRowItemID.
- If @tp_ID is NULL and @BaseRowItemID is NULL, proc_AddListItem MUST generate a new value for the list item identifier that is unique within the list (1).

@UserId: The user identifier for the current user. proc_AddListItem uses this for purposes of permission-checking. This value MUST refer to an existing user identifier for the specified site collection.

@Size: The size in bytes of the list item row to be added. This parameter MUST NOT be NULL.

@TimeNow: The current time, in UTC format, on the back-end database server.

@CopySecurityFromMasterID: Specifies the optional identifier of the list item to copy the security scope settings from for this list item. A list item that represents an exception to a recurrence item in a Meetings List (a list (1) with a List Server Template type value (section 2.2.1.2.12) of 200) MUST have the same security scope settings as the master recurrence item. If this parameter is set to a master recurrence item's list item identifier and the list item to be added does not share the security scope of the master recurrence item, the new list item MUST be given a unique security scope with a copy of the security settings from the security scope of the master recurrence item. This parameter MUST only be set for list items that are exceptions to recurrence items in a Meetings List.

@ExtraItemSize: The size of the predefined SQL parameter fields in the list item row being added.

@CheckDiskQuota: A bit flag specifying whether or not Disk Quota is checked for the current user before adding the list item. The bit MUST be set to one for Disk Quota to be checked for the current user before adding the list item; otherwise, the bit MUST be set to zero.

@ItemDocType: The Document Store Type (section 2.2.4) of the list item being added to the list (1).
@SortTypeReversed: Specifies whether or not the list item sorts using the behavior of folder or file types. The bit MUST be set to one for folders to sort like files; otherwise, it MUST be set to zero. It MUST be set to zero for all file type list items.

@BaseRowItemId: An optional value that specifies the list item identifier to be added if a value is not supplied by the @tp_Id parameter. If the @tp_Id parameter is NULL and this parameter is not NULL, proc_AddListItem MUST use this value for the list item identifier to be added.

@DocIdAdded: The optional document identifier of the document to be inserted as a list item in the list (1). If @DocIdAdded is not NULL and the document has a document identifier, this parameter MUST be the existing document identifier. If this parameter is NULL, a new document identifier MUST be generated for the list item. If this list item is being restored to the list (1) as part of an implementation-specific backup restore operation, as specified by the value of @RetainObjectIdentity, this parameter MUST be ignored and the value of the @RetainId parameter MUST be used as the document identifier.

@RetainId: The document identifier of the document to be inserted as a list item in the list (1) if this list item is being restored to the list (1) as part of a back-up restore operation.

@RetainObjectIdentity: A bit flag specifying whether this list item is being restored to the list (1) as part of an implementation-specific backup restore operation. If @RetainObjectIdentity is set to one, this list item is being restored to the list (1) as part of an implementation-specific backup restore operation and MUST have the @DocIdAdded value specified in @RetainId.

@Level: The Publishing Level Type value (section 2.2.6) specifying the publishing level of this list item.

@UIVersion: The UI version number for the list item.

@ItemCountDelta: The number to be added to the list item count of the containing list (1):

- For a list item added with a single call to proc_AddListItem or for one call for a list item with multiple rows to be added, this value MUST be one.
- For the other calls to proc_AddListItem for the additional rows to be added for the list item, this value MUST be zero.

@ItemName: The display name of the list item.

@UseNvarchar1ItemName: If @ItemName is NULL, this bit flag specifies whether to use the content of @nvarchar1 for the display name of the list item.

@ItemDirName: An output parameter containing the directory name of the list item.

@ItemLeafName: An output parameter containing the leaf name of the list item.

@ServerTemplate: The identifier for the List Server Template defining the base structure of the list (1) containing this list item.

@IsNotUserDisplayed: A bit flag specifying whether the user name is not displayed with list items.

@BaseType: The List Base Type (section 2.2.1.11) of the list (1) containing the list item.

@CheckSchemaVersion: Specifies an optional schema version number to compare with the list schema version number. If this parameter is not NULL, the version numbers MUST match for successful completion.

@OnRestore: A bit flag that specifies whether this list item is being inserted by an implementation-specific back-up restore operation.
@AddNamespace: A Boolean value specifying whether metadata is being added to the list item. This parameter MUST NOT be NULL.

@tp_Ordering: Specifies the threading structure for this list item in a deprecated discussion board list (1) (a list (1) with a List Base Type of three) as a concatenation of UTC date/time stamp values in yyyyMMddHHmmss format. For all list items in lists (1) with other List Base Types, this parameter MUST be NULL.

@tp_ThreadIndex: Specifies the list item position within a threaded discussion board list (1) (a list (1) with a List Base Type of three) as a binary structure. For all list items in lists (1) with other List Base Types, this parameter MUST be NULL.

@tp_HasAttachment: A bit flag that specifies whether the list item has an associated attachment.

@tp_ModerationStatus: A Moderation Status (section 2.2.1.2.13) value specifying the current moderation status of this list item.

@tp_IsCurrent: A bit flag that specifies whether or not this is the current version of this publishing level of the list item.

@tp_ItemOrder: Specifies the implementation-specific order in which the list item is displayed with other list items from the same list (1). This value can be the same as other list items in the list (1).

@tp_InstanceID: Specifies a meeting instance identifier if the list item is associated with the particular meeting instance of a recurring meeting. For all other list items, this parameter MUST be NULL.

@tp_GUID: A GUID that uniquely identifies the list item within the AllUserData table.

@tp_Id: The optional List Item Identifier (section 2.2.1.1.6) specified for this list item. If this parameter is not NULL, proc_AddListItem MUST use this value for the identifier of the list item to be added.

@tp_Author: The user identifier for the user who created the list item.

@tp_Editor: The user identifier for the user who last edited the list item.

@tp_Modified: A time stamp, in UTC format, that specifies when this list item was last modified.

@tp_Created: A time stamp, in UTC format, that specifies when this list item was created.

@tp_Version: Specifies the internal version number used for internal conflict detection.

@tp_ContentTypeId: Specifies the content type identifier for this list item.

@tp_CopySource: Specifies the URL used as a source for this list item. If this list item was not copied from a source list item, this value MUST be NULL.

@tp_HasCopyDestinations: A bit flag specifying whether destination locations have been set for this list item to be copied to. If this list item does not have a destination location set, this value MUST be zero.

@tp_WorkflowVersion: If the list item is part of a workflow, this parameter specifies the value to set denoting the state of the list item within that workflow. If the list item is not part of a workflow, this MUST be NULL.

@tp_WorkflowInstanceID: A workflow identifier that specifies the currently active workflow instance on this list item. If this list item is not part of a workflow, this MUST be NULL.

@nvarchar1: User-defined column in the list (1) containing a value of type nvarchar. If the column contains no data, the value MUST be NULL.
@MtgEventType: If @ServerTemplate corresponds to a Meeting Workspace site, this parameter specifies the type of calendar event.

@EventUID: If @ServerTemplate corresponds to a Meeting Workspace site, this parameter specifies the unique identifier for the meeting instance.

@RecurrenceID: If @ServerTemplate corresponds to a Meeting Workspace site, this parameter specifies a particular instance of a recurring meeting.

@tp_ColumnSet: The XML fragment that specifies the values of user-defined columns in the list (1). The name of each element specified in this parameter MUST be one of the columns defined in the AllUserData table with the SPARSE keyword.

@eventData: Contains implementation-specific event (1) data that is significant to the front-end Web server but is otherwise opaque to the back-end database server. To be stored by the BEDS for eventual writing to a log file.

@acl: The binary serialization of the WSS ACL Format (section 2.2.3.6) ACL for the data supplied in @eventData. To be stored with the data.

@DocClientId: An optional parameter that specifies the client identifier of the list item to be saved. This can be NULL.

@tp_AppAuthor: The identifier of the app principal who created the list item.

@tp_AppEditor: The app principal identifier of the app principal who last edited the list item.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_AddListItem stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>2</td>
<td>Postprocessing of the list item failed because a prerequisite list item was not found.</td>
</tr>
<tr>
<td>3</td>
<td>The directory specified for the list item does not exist.</td>
</tr>
<tr>
<td>5</td>
<td>The attempt to create a directory or document failed because the user does not have sufficient permissions.</td>
</tr>
<tr>
<td>13</td>
<td>The list item to be added is not valid.</td>
</tr>
<tr>
<td>16</td>
<td>Adding the list item caused updating of an existing list item to fail.</td>
</tr>
<tr>
<td>33</td>
<td>Cannot move directories that contain checked-out files.</td>
</tr>
<tr>
<td>50</td>
<td>A list item could not be deleted.</td>
</tr>
<tr>
<td>80</td>
<td>The document being added to the list already exists.</td>
</tr>
<tr>
<td>87</td>
<td>Unable to add the list item because the input parameters do not match existing list items, or an error occurred during a table update operation.</td>
</tr>
<tr>
<td>138</td>
<td>The list could not be moved to the specified location.</td>
</tr>
<tr>
<td>160</td>
<td>Could not create a unique filename.</td>
</tr>
<tr>
<td>161</td>
<td>A directory that spans sites cannot be moved.</td>
</tr>
<tr>
<td>183</td>
<td>The list item being added already exists in the list.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>206</td>
<td>The file or directory name is too long.</td>
</tr>
<tr>
<td>212</td>
<td>The database for the site collection is locked.</td>
</tr>
<tr>
<td>1150</td>
<td>Failed to update the list.</td>
</tr>
<tr>
<td>1359</td>
<td>An internal error occurred while moving a list item.</td>
</tr>
<tr>
<td>1638</td>
<td>The current schema version of the list does not match the value in @CheckSchemaVersion.</td>
</tr>
<tr>
<td>1816</td>
<td>The site collection is over its allocated size quota.</td>
</tr>
<tr>
<td>8398</td>
<td>A directory could not be deleted.</td>
</tr>
</tbody>
</table>

The **proc_AddListItem** stored procedure MUST return no result sets.

### 3.1.5.5 proc_ChangeLevelForDoc

The **proc_ChangeLevelForDoc** stored procedure is invoked to update the publishing level of a document. Depending on the settings in effect on the containing list and the permissions of the specified user making the request, this procedure also can update the document's moderation status, UI version number, and properties to reflect the change.

```sql
PROCEDURE proc_ChangeLevelForDoc(  @SiteId                        uniqueidentifier,  @DirName                       nvarchar(256),  @LeafName                      nvarchar(128),  @Level                         tinyint                     OUTPUT,  @NewLevel                       tinyint,  @ModerationStatus              int,  @EnableMinorVersions           bit,  @Moderated                      bit,  @CreateVersion                 bit,  @UserId                        int,  @Comment                       nvarchar(1023),  @bUpdateModified               bit,  @DoclibRowId                   int                         OUTPUT,  @DocUIVersion                 int                         OUTPUT,  @DocFlagsOut                   int                         OUTPUT,  @DocVersionOut                  int                         OUTPUT,  @RequestGuid                   uniqueidentifier   = NULL   OUTPUT);
```

**@SiteId**: The site collection identifier of the site collection containing the document.

**@DirName**: The directory name of the document.

**@LeafName**: The leaf name of the document.

**@Level**: A Publishing Level Type (section 2.2.2.6) value indicating the document's current publishing level. This output parameter also reflects the new publishing level value for the document.

**@NewLevel**: A Publishing Level Type (section 2.2.2.6) value indicating the requested publishing level for the document. If the document is currently checked out and **@NewLevel** is set to 1 (Published), then the document's level value MUST be changed to 1 (Published). Otherwise, the document's level value MUST be changed to 255 (Checked Out). If **@NewLevel** is set to 255 (Checked Out), **@EnableMinorVersions** is 0 and **@NewLevel** is set to 1 (Published), then the document's level value MUST NOT change, and output parameters MUST NOT be changed. If the document's current Level is 255 (Checked Out) and the **@NewLevel** is 2 (Draft), the document's **CheckOutUserId**, **CheckOutDate** and **CheckOutUser** fields MUST be updated to reflect the new level value.
CheckoutDate and CheckoutExpires values MUST be set to NULL, and the document's container (document library or list) value for tp_CheckOutUserId MUST be set to NULL.

@ModerationStatus: An integer value indicating the specified document's Moderation Status (section 2.2.1.2.13). See the Moderation Status section for a list of all valid values.

@EnableMinorVersions: A bit indicating whether minor versions are enabled within the containing list. If the list containing the specified document has minor versions enabled, then this MUST be set to 1; otherwise, this MUST be set to 0.

@Moderated: A bit indicating whether moderation is enabled on the containing list. If the list containing the specified document has moderation enabled, then this MUST be set to 1; otherwise, this MUST be set to 0.

@CreateVersion: A bit indicating whether versioning is enabled on the containing list. If the list containing the specified document has versioning enabled, then this MUST be set to 1; otherwise, this MUST be set to 0.

@UserId: A user identifier for the user requesting the document change. This value MUST refer to an existing User Identifier for the specified site collection.

@Comment: Descriptive text associated with the document on check in or publishing. When the document publishing level is updated from "Check Out" (255) to any other value, or from any value to "Publish" (1), @Comment can be set to a descriptive string or it can be NULL. In all other Publishing Level Type value transitions, this parameter MUST be NULL.

@bUpdateModified: A bit indicating if change to the last modified date and time properties of the document is requested. If @bUpdateModified is set to 1, then proc_ChangeLevelForDoc MUST update document last modified date and time properties in the store to note that the document was updated at the current time. Otherwise, the last modified date and time properties MUST be left unchanged.

@DoclibRowId: An output parameter containing the value of the DoclibRowId column in the Docs View (section 2.2.6.3) for the document after the change.

@DocUIVersion: An output parameter containing the UI version of the document after the change.

@DocFlagsOut: An output parameter containing the Doc Flags (section 2.2.2.3) of the document after the change.

@DocVersionOut: An output parameter containing the internal version number of the document after the change.

@RequestGuid: The optional request identifier for the current request.

Return Values: This procedure returns an integer return code, which MUST be included in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>87</td>
<td>The specified document could not be found, or the specified publishing level is not the current level of the specified document.</td>
</tr>
</tbody>
</table>

This procedure MUST NOT return a result set.
3.1.5.6 proc_CheckRbsInstalled

The proc_CheckRbsInstalled stored procedure is invoked to test whether the back-end database server can support remote BLOB storage.

```
PROCEDURE proc_CheckRbsInstalled(  
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_CheckRbsInstalled stored procedure returns an integer return code, which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The back-end database server does not support remote BLOB storage.</td>
</tr>
<tr>
<td>1</td>
<td>The back-end database server supports remote BLOB storage.</td>
</tr>
</tbody>
</table>

The proc_CheckRbsInstalled stored procedure MUST NOT return any result sets.

3.1.5.7 proc_CheckoutDocument

The proc_CheckoutDocument stored procedure is invoked to place a short-term lock on a document, refresh, convert or release an existing short-term lock, or to check out a document. Short term locks are either of type shared or exclusive.

```
PROCEDURE proc_CheckoutDocument(
    @SiteId uniqueidentifier,
    @WebId uniqueidentifier,
    @DirName nvarchar(256),
    @LeafName nvarchar(128),
    @Level tinyint,
    @EnableMinorVersions bit,
    @IsModerated bit,
    @UserId int,
    @CheckoutTimeout int,
    @RefreshLock bit,
    @CheckoutToLocal bit,
    @IsForceCheckout bit,
    @IsSharedLock bit,
    @IsConvertLock bit,
    @DocMetaInfo varbinary(max),
    @DocMetaInfoSize int,
    @DocMetaInfoVersion int,
    @GetLinkInfo bit,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier for the site collection containing the document to be checked out, locked, or unlocked.

@WebId: The site identifier for the site containing the document.

@DirName: The directory name of the document.

@LeafName: The leaf name of the document.
@Level: A Publishing Level Type (section 2.2.6) value specifying the publishing status of the document.

@EnableMinorVersions: A bit flag specifying whether the document library or list containing the document has minor version numbering enabled. If this parameter is set to "1", then minor version numbering is enabled for the document library or list; otherwise, this MUST be set to 0.

@IsModerated: A bit flag specifying whether moderation is in effect on the document. If this parameter is set to "1", then moderation is in effect on this document, which is used to implement an approval process to set the publishing level to published after the document is created or modified. This parameter MUST NOT be NULL.

@UserId: The User Identifier (section 2.2.1.1.13) for the current user who is requesting a short-term lock or checking out the document. This value MUST refer to an existing user identifier for the specified site collection.

@CheckoutTimeout: Specifies the remaining time in minutes that short-term locking will be in effect for the document. A value of 0 means that the existing short-term lock MUST be released. The @CheckoutTimeout parameter MUST be NULL if the document is being checked out instead of having a short-term lock applied.

@RefreshLock: A bit flag specifying whether the short-term lock on the document is being refreshed. If this parameter is set to "1", then the existing short-term lock on the document MUST be refreshed for the number of minutes specified by the @CheckoutTimeout parameter. This parameter MUST be set to "0" to check out the document or to request a new short-term lock. This parameter MUST NOT be NULL.

@CheckoutToLocal: A bit flag specifying whether the document is to be copied to local storage on the user's client for editing. If this parameter is set to "1", then the user's client MUST make a local copy of the document stream for editing, and proc_CheckoutDocument MUST NOT make a checked-out version of the document in the store. This parameter MUST NOT be NULL.

@IsForceCheckout: A bit flag specifying whether the containing document library requires documents to be checked out before any changes can be made. If this parameter is set to "1", then the containing document library has 'Require Check Out' turned on. This parameter MUST NOT be NULL.

@IsSharedLock: A bit flag specifying whether the desired short term lock on the document is a shared lock or an exclusive lock. This parameter MUST be set to "1" if the desired short term lock is type shared; otherwise it MUST be set to "0".

@IsConvertLock: A bit flag specifying whether to convert an existing short term lock from one short term lock type to a different short term lock type. This parameter MUST be set to "1" to convert the type of an existing short term lock; otherwise it MUST be set to "0".

@DocMetaInfo: The metadata information for the document to be checked out. If there is no metadata information for this document, then this parameter MUST be NULL.

@DocMetaInfoSize: Size in bytes of the document's metadata info. This MUST be NULL if @DocMetaInfo is NULL.

@DocMetaInfoVersion: The version of the metadata information for the document to be checked out. This MUST be NULL if @DocMetaInfo is NULL.

@GetLinkInfo: A bit flag specifying whether the desired Link Info Single Doc Result Set, specified in section 3.1.5.7.1, is returned. This parameter MUST be set to "1" to return the Link Info Single Doc Result Set; otherwise it MUST be set to "0".

@RequestGuid: The optional request identifier for the current request.
Return Values: The `proc_CheckoutDocument` stored procedure returns an integer return code, which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>File not found. A document corresponding to the specified <code>@SiteId</code>, <code>@WebId</code>, <code>@DirName</code>, <code>@LeafName</code>, and <code>@Level</code> parameters was not found.</td>
</tr>
<tr>
<td>33</td>
<td>Short-term lock error. The document cannot have a short-term lock applied because another user has the document checked out.</td>
</tr>
<tr>
<td>154</td>
<td>Invalid minor version value. The minor version value for the document would exceed the maximum allowed value (511) if checked out.</td>
</tr>
<tr>
<td>158</td>
<td>Checkout required. The document is in a document library with the &quot;Require Check Out&quot; option set, but the document is not being checked out.</td>
</tr>
<tr>
<td>173</td>
<td>Checkout error. The document cannot be checked out, because it is already checked out to or locked by another user.</td>
</tr>
<tr>
<td>212</td>
<td>Site collection locked. The site collection is in disk write lock.</td>
</tr>
<tr>
<td>1630</td>
<td>Unsupported document type. The document specified is not valid for check out; folders and sites cannot be checked out.</td>
</tr>
<tr>
<td>1816</td>
<td>Disk quota error. The site collection disk quota has been reached.</td>
</tr>
<tr>
<td>6002</td>
<td>Short-term lock error. The document cannot have a short-term lock applied because another user has a shared short-term lock on the file. A shared short-term lock is used for read operations that do not change or update data, such as a SELECT statement.</td>
</tr>
<tr>
<td>6009</td>
<td>Short-term lock error. The document cannot have a short-term lock applied because another user has an exclusive short-term lock on the file. An exclusive short-term lock is used for data-modification operations, such as INSERT, UPDATE, or DELETE. Such a lock ensures that multiple updates cannot be made to the same resource at the same time.</td>
</tr>
</tbody>
</table>

The `proc_CheckoutDocument` stored procedure MUST return multiple result sets. Some of the result sets are returned 0 or more times depending upon conditions specified in the following sections, and all result sets that are returned will be sent in the order specified in the sections 3.1.5.7.1 through 3.1.5.7.4.

### 3.1.5.7.1 Link Info Single Doc Result Set

The `Link Info Single Doc Result Set` returns information about all forward links and backward links associated with the document. The `Link Info Single Doc Result Set` MUST be returned once and MUST hold one row for each forward and backward link associated with the specified document at the specified publishing level.

The `Link Info Single Doc Result Set` is defined in the Common Result Sets `Link Information Result Set` (section 2.2.4.13).

### 3.1.5.7.2 Document Metadata Result Set

The `Document Metadata Result Set` returns the metadata for the document. The `Document Metadata Result Set` MUST be returned and MUST contain a single row corresponding to the checked-out or locked document.

The `Document Metadata Result Set` is defined in the Common Result Sets `Document Metadata Result Set` (section 2.2.4.8).
3.1.5.7.3 Event Receivers Result Set

The Event Receivers Result Set contains information about the list item event receivers defined for this document.

The Event Receivers Result Set MUST only be returned if the requested document matching the @Level parameter exists within the site (2) specified by the @WebId parameter for the current user. The Event Receivers Result Set MUST contain one row for each event receiver registered with an Event Host Type (section 2.2.1.2.5) of 3 (list item) for this document.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.4.11).

3.1.5.7.4 Audit Mask Result Set

The Audit Mask Result Set returns audit configuration information for the document. The Audit Mask Result Set MUST be returned with one row of audit configuration information on successful execution.

The Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).

3.1.5.8 proc_ClearLinks

The proc_ClearLinks stored procedure deletes all link information associated with a particular field in a list item as part of a list item or document update. All link information that matches all of the parameters and does not have associated Web Parts MUST be deleted by proc_ClearLinks. Link information associated with Web Parts MUST NOT be deleted by proc_ClearLinks.

```sql
PROCEDURE proc_ClearLinks(
    @SiteId                        uniqueidentifier,
    @DirName                       nvarchar(256),
    @LeafName                      nvarchar(128),
    @Level                         tinyint,
    @FieldId                       uniqueidentifier,
    @RequestGuid                   uniqueidentifier = NULL   OUTPUT
);
```

@SiteId: The site collection identifier of the site collection that contains the list item with the link information to be deleted.

@DirName: The directory name of the list item with the link information to be deleted.

@LeafName: The leaf name of the list item with the link information to be deleted.

@Level: The Publishing Level Type (section 2.2.2.6) value associated with the list item with the link information to be deleted.

@FieldId: The field identifier of the field in the list item with the link information to be deleted.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_ClearLinks stored procedure returns an integer return code, which MUST be 0. The proc_ClearLinks stored procedure MUST NOT return a result set.

3.1.5.9 proc_CreateDir

The proc_CreateDir stored procedure is invoked to create a directory or folder with a specified name at a specified location.
PROCEDURE proc_CreateDir(
    @DirSiteId uniqueidentifier,
    @DirWebId uniqueidentifier,
    @DirDirName nvarchar(256) OUTPUT,
    @DirLeafName nvarchar(128) OUTPUT,
    @DirLevel tinyint,
    @AddMinorVersion bit,
    @DocFlags int,
    @CreateDirFlags int,
    @UserId int = NULL,
    @AppPrincipalId int = NULL,
    @ProgId nvarchar(255) = NULL,
    @DirMetaInfo varbinary(max) = NULL,
    @DirMetaInfoSize int = 0,
    @DirClientId uniqueidentifier = NULL OUTPUT,
    @ScopeId uniqueidentifier = NULL OUTPUT,
    @DoclibRowIdRequired int = NULL,
    @ReverseSortOrder bit = 0,
    @ScopeIdOverride uniqueidentifier = NULL,
    @bAlreadyExists bit = NULL OUTPUT,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);

@DirSiteId: The site collection identifier (section 2.2.1.1.9) for the site collection containing the directory to be created.

@DirWebId: The site identifier for the site (2) containing the directory to be created.

@DirDirName: This is both an input and an output parameter. The input parameter MUST specify the store-relative form URL of the parent location in which the specified directory is to be created. The directory name of the created directory MUST be returned as the output parameter value.

@DirLeafName: This is both an input and an output parameter. The input parameter MUST specify the name of the directory to be created in the parent location specified by the @DirDirName parameter. The leaf name of the created directory MUST be returned as the output parameter value.

@DirLevel: A Publishing Level Type value (section 2.2.2.6) specifying the publishing status of the directory to be created. This value MUST be one (Published) if the directory is not being created in a list (1) or document library. If this parameter is two, specifying that the list item is to be created as a draft, the @UserId value MUST be set as the draft owner for the created directory.

@AddMinorVersion: If this parameter has a value of one, proc_CreateDir MUST set the major version number of the created directory to zero and the minor version number to one. If this parameter has a value of zero, proc_CreateDir MUST set the major version number of the created directory to one and the minor version number to zero. This parameter MUST NOT be NULL.

@DocFlags: A Doc Flags value (section 2.2.2.3) specifying metadata about the directory to be created. If the @DocFlags value has the 0x00002000 bit set (specifying a custom ordering of content types), this bit flag MUST be ignored and MUST NOT be included in the metadata associated with the created directory.

@CreateDirFlags: A 4-byte unsigned integer bit mask containing options used while creating the directory. The value MUST be zero, with none of the bits set, or bits MUST be set as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000007</td>
<td>These three bits contain an Attachment Flag value (section 2.2.1.2.1).</td>
</tr>
<tr>
<td>0x00000008</td>
<td>Return an access denied error if the specified directory already exists.</td>
</tr>
<tr>
<td>0x000000010</td>
<td>Do not promote the directory to a document library.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x00000020</td>
<td>The directory contains a thicket.</td>
</tr>
<tr>
<td>0x00000380</td>
<td>These three bits contain a Moderation Status value (section 2.2.1.2.13).</td>
</tr>
<tr>
<td>0xFFFFFC40</td>
<td>These values are currently unused and MUST be ignored.</td>
</tr>
</tbody>
</table>

@UserId: The User Identifier (section 2.2.1.1.13) for the current user who is directly or indirectly requesting the directory creation. If this parameter is not NULL, it MUST be used by proc_CreateDir for permission checking and setting ownership of the created directory. If this parameter is NULL, the ownership of the created directory MUST be set from the parent.

@AppPrincipalId: The app principal identifier of the app that is requesting this operation. If the operation was not requested by an app then it MUST be set to 0.

@ProgId: An optional parameter that specifies a preferred application to open the directory. The @ProgId value is used to distinguish between different applications that save files with a given file extension. This can be NULL.

@DirMetaInfo: The metadata information for the directory to be created. This can be NULL.

@DirMetaInfoSize: Size, in bytes, of the directory's metadata info. This MUST be zero if @DirMetaInfo is NULL.

@DirClientId: An optional parameter that specifies the client identifier of the directory to be created. This can be NULL.

@DirId: The identifier for the created directory. This is both an input and output parameter. If this parameter is passed in to proc_CreateDir with a non-NULL value, this value MUST be unique in the back-end database server and MUST be used as the document identifier for the directory to be created. If this parameter is passed in with a NULL value, a new uniquedidentifier value MUST be generated for the directory. If proc_CreateDir creates a new directory, it MUST return the document identifier for the created directory in the output parameter upon successful completion. If proc_CreateDir does not create a new directory (for example, if the directory already exists), this output parameter MUST be ignored.

@ScopeId: An output parameter that MUST contain the non-NULL value specified for @ScopeIdOverride or if a NULL value is specified for @ScopeIdOverride, the Scope Identifier (section 2.2.1.1.8) of the parent directory. This identifies the specific ACL to use for calculating the permission settings on the created directory.

@DoclibRowIdRequired: Specifies the List Item Identifier (section 2.2.1.1.6) to be used for this directory if created within a document library. If this parameter is not NULL and the directory to be created is within a document library, proc_CreateDir MUST set this as the list item identifier for the created directory. This parameter MUST be NULL for directories that are not created within a document library.

@ReverseSortOrder: Specifies whether the directory sorts using the behavior of the file or directory type. The bit MUST be set to "1" for directories to sort like files; otherwise, it MUST be set to "0".

@ScopeIdOverride: Specifies the scope to use for the directory to be created. If this parameter is not NULL, it MUST be set as the Scope Identifier (section 2.2.1.1.8) for the directory. Otherwise, the Scope Identifier of the parent directory MUST be used. The scope set for the created directory MUST be returned in the @ScopeId output parameter.

@bAlreadyExists: An output parameter containing a bit flag set to zero if the directory did not already exist and set to one if the directory already existed.

@RequestGuid: The optional request identifier for the current request.
Return Values: The `proc_CreateDir` stored procedure MUST return an integer return code that MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The parent directory was not found.</td>
</tr>
<tr>
<td>5</td>
<td>The current user does not have sufficient permissions to create the directory at the specified location.</td>
</tr>
<tr>
<td>13</td>
<td>The list item to be added is not valid.</td>
</tr>
<tr>
<td>16</td>
<td>Adding the list item caused updating of an existing list item to fail.</td>
</tr>
<tr>
<td>80</td>
<td>The specified directory already exists.</td>
</tr>
<tr>
<td>87</td>
<td>Unable to add the list item because the input parameters do not match existing list items, or an error occurred during a table update operation.</td>
</tr>
<tr>
<td>206</td>
<td>The file or directory name is too long.</td>
</tr>
<tr>
<td>212</td>
<td>The site collection is locked.</td>
</tr>
<tr>
<td>1150</td>
<td>Failed to update the list (1).</td>
</tr>
<tr>
<td>1816</td>
<td>There is not enough database quota for the current user to complete the operation.</td>
</tr>
</tbody>
</table>

The `proc_CreateDir` stored procedure MUST NOT return any result sets.

### 3.1.5.10 proc_DeleteAllDocumentVersions

The `proc_DeleteAllDocumentVersions` stored procedure is invoked to delete either the draft versions or older published versions of a document and optionally place them in the recycle bin.

```sql
PROCEDURE proc_DeleteAllDocumentVersions(
    @DocSiteId                     uniqueidentifier,
    @DocDirName  nvarchar(256),
    @DocLeafName nvarchar(128),
    @UserId            int,
    @AppPrincipalId    int,
    @DeleteOp          int,
    @Level             tinyint,
    @RequestGuid       uniqueidentifier   = NULL      OUTPUT
);
```

- **@DocSiteId**: The site collection identifier (section 2.2.1.1.9) of the site collection containing the document whose specified versions are being deleted.

- **@DocDirName**: The directory name of the document whose specified versions will be deleted. If there is no forward slash in the directory name, then the value MUST be an empty string.

- **@DocLeafName**: The leaf name of the document whose specified versions will be deleted.

- **@UserId**: The user identifier (section 2.2.1.1.13) of the current user requesting the deletion. This value MUST refer to an existing user identifier for the site collection specified by `@DocSiteId`.

- **@AppPrincipalId**: The app principal identifier of the app that is requesting this operation. If the request if not made by an app then it MUST be set to 0.
@DeleteOp: A parameter specifying the delete options. The value MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>The deleted document versions MUST NOT be placed in the recycle bin (nonrecoverable delete).</td>
</tr>
<tr>
<td>4</td>
<td>The deleted document versions MUST be placed in the recycle bin (recoverable delete).</td>
</tr>
</tbody>
</table>

@Level: A Publishing Level Type (section 2.2.2.6) value specifying which versions of the document are to be deleted. The value MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All versions of the document MUST be deleted, excluding the current version and the current published version. If the current version and the current published version are the same, then only one version will remain.</td>
</tr>
<tr>
<td>2</td>
<td>All draft versions of the document MUST be deleted, excluding the current version.</td>
</tr>
</tbody>
</table>

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_DeleteAllDocumentVersions stored procedure returns an integer return code that MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>2</td>
<td>The document specified cannot be found in the site collection.</td>
</tr>
<tr>
<td>1359</td>
<td>Invalid Parameter: @DeleteOp does not contain a valid value.</td>
</tr>
</tbody>
</table>

The proc_DeleteAllDocumentVersions stored procedure MUST NOT return any result sets.

3.1.5.11   proc_DeleteDocBuildDependencySet

The proc_DeleteDocBuildDependencySet stored procedure is invoked to delete a build dependency set for a specified document.

```sql
PROCEDURE proc_DeleteDocBuildDependencySet(
    @DocSiteId                     uniqueidentifier,
    @DocDirName                    nvarchar(256),
    @DocLeafName                   nvarchar(128),
    @Level                         tinyint,
    @RequestGuid                   uniqueidentifier   = NULL      OUTPUT
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection containing the document.

@DocDirName: The directory name containing the document. If there is no forward slash in the directory name, the value MUST be an empty string.

@DocLeafName: The leaf name containing the document.

@Level: The Publishing Level Type (section 2.2.2.6) value of the document.

@RequestGuid: The optional request identifier for the current request.
Return Values: The `proc_DeleteDocBuildDependencySet` stored procedure MUST return an integer return code of 0.

The `proc_DeleteDocBuildDependencySet` stored procedure MUST NOT return a result set.

### 3.1.5.12 proc_DeleteDocumentVersion

The `proc_DeleteDocumentVersion` stored procedure is invoked to delete a document version and optionally place it in the recycle bin. A current version (whether published or draft) cannot be deleted. A current version is either the current published version or the current draft version of the document.

```sql
PROCEDURE proc_DeleteDocumentVersion(  
    @DocSiteId   uniqueidentifier,  
    @DocDirName  nvarchar(256),  
    @DocLeafName nvarchar(128),  
    @DocVersion  int,  
    @UserId      int,  
    @AppPrincipalId int,  
    @DeleteOp    int,  
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@DocSiteId: The site collection identifier (section 2.2.1.1.9) of the site collection containing the document whose specified version is being deleted.

@DocDirName: The directory name of the document to delete. If there is no forward slash in the directory name, then the value MUST be an empty string.

@DocLeafName: The leaf name of the document to delete.

@DocVersion: The version of the document to delete. This value MUST NOT be a current version.

@UserId: The User Identifier (section 2.2.1.1.13) of the current user requesting the deletion. This value MUST refer to an existing user identifier for the specified site collection.

@AppPrincipalId: The app principal identifier of the app requesting the deletion. If the request is not made by an app then it MUST be set to 0.

@DeleteOp: A value determining delete options. The value MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>The deleted document version MUST NOT be placed in the recycle bin (non-recoverable delete).</td>
</tr>
<tr>
<td>4</td>
<td>The deleted document version MUST be placed in the recycle bin (recoverable delete).</td>
</tr>
</tbody>
</table>

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_DeleteDocumentVersion` stored procedure returns an integer return code that MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>2</td>
<td>The document version specified by @DocSiteId, @DocDirName, @DocLeafName and @DocVersion cannot be found.</td>
</tr>
<tr>
<td>186</td>
<td>Invalid Parameter: An invalid version was specified. The value specified in @DocVersion is a current version.</td>
</tr>
</tbody>
</table>
The `proc_DeleteDocumentVersion` stored procedure MUST NOT return any result sets.

### 3.1.5.13 `proc_DeleteUrl`

The `proc_DeleteUrl` stored procedure is invoked to delete a document. `proc_DeleteUrl` accepts documents of all types except sites or attachment folders.

```sql
PROCEDURE proc_DeleteUrl(
  @WebSiteId uniqueidentifier,
  @WebId uniqueidentifier,
  @Url nvarchar(260),
  @UserId int,
  @AppPrincipalId int,
  @LogChange bit = 1,
  @ListDeletedUrls bit = 1,
  @ListDeletedAliases bit = 0,
  @AttachmentsFlag tinyint = 0,
  @AttachmentOp int = 3,
  @IgnoreCheckedOutFiles bit = 0,
  @IgnoreLookupRelationshipsCheck bit = 0,
  @DeleteOp int = 3,
  @ThresholdRowCount int = 0,
  @QueryAuditFlags bit = 0,
  @FailedUrl nvarchar(260) = NULL OUTPUT,
  @DeleteTransactionId uniqueidentifier = '00000000-0000-0000-0000-000000000000' OUTPUT,
  @eventData varbinary(max) = NULL,
  @acl varbinary(max) = NULL,
  @IsCascadeDeleteOperation bit = 0,
  @IsCascadeParent bit = 0,
  @ChildDeleteTransactionId varbinary(16) = NULL OUTPUT,
  @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@WebSiteId: The site collection identifier (section 2.2.1.1.9) of the site collection containing the document.

@WebId: The site identifier (section 2.2.1.1.11) of the site (2) containing the document.

@Url: The store-relative form URL of the document.

@UserId: The user identifier (section 2.2.1.1.13) for the current user requesting the operation. This parameter is used by `proc_DeleteUrl` for purposes of permission-checking. This value MUST refer to an existing user identifier for the specified site collection.

@AppPrincipalId: The app principal identifier of the app requesting this operation. If this operation was not requested by an app then it MUST be set to 0.

@LogChange: This parameter is reserved for internal use and MUST be the default value.

@ListDeletedUrls: If this bit is set to 1, then the result set information about the deleted documents is requested. See the Deleted Documents Result Set (section 3.1.5.13.1).

@ListDeletedAliases: If this bit is set to 1, then the result set information about deleted lists that had configured e-mail addresses is requested. See the Deleted Aliased Lists Result Set (section 3.1.5.13.2).

@AttachmentsFlag: An Attachments Flag (section 2.2.1.2.1) describing the document specified by @Url.
@AttachmentOp: This parameter specifies whether or not the attachment folder record in the store MUST be updated to reflect whether or not it has any attachments remaining, and which metadata in the store for the folder MUST be refreshed, as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No updates.</td>
</tr>
<tr>
<td>1</td>
<td>The attachment folder record in the store MUST be updated to reflect whether it has any attachments remaining.</td>
</tr>
<tr>
<td>2</td>
<td>The attachment folder record in the store MUST be updated, and the internal version number of the folder MUST be incremented.</td>
</tr>
<tr>
<td>3</td>
<td>The attachment folder record in the store MUST be updated, and the last modified date and time of the folder MUST be refreshed.</td>
</tr>
</tbody>
</table>

@IgnoreCheckedOutFiles: This parameter is reserved for internal use and MUST be the default value.

@IgnoreLookupRelationshipsCheck: This parameter is reserved for internal use and MUST be the default value.

@DeleteOp: This value defines the type of delete operation to attempt. If no value is specified, then a default value of 3 is used. Otherwise, @DeleteOp MUST be one of the values listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Delete the item without placing the deleted item into the recycle bin.</td>
</tr>
<tr>
<td>4</td>
<td>Delete the item and place the deleted item into the recycle bin.</td>
</tr>
</tbody>
</table>

@ThresholdRowCount: If this value is not 0, then the stored procedure MUST NOT proceed with the delete operation if the number of document rows deleted by the operation exceeds the value of this parameter.

@QueryAuditFlags: This parameter is set to 1 to specify that an audit entry MUST be created for the delete operation.

@FailedUrl: An output parameter indicating the store-relative form URL at which the delete operation failed. This parameter MUST be set to NULL if the deletion was successful.

@DeleteTransactionId: This parameter is a value which is used to identify all files and items that are deleted as part of a single SQL transaction. The first call to proc_DeleteUrl within the SQL transaction MUST pass NULL. The implementation assigns a new delete transaction ID and returns it as output. On output, this parameter MUST return either an all-zero GUID, indicating that the item was not placed into the recycle bin, or a valid transaction ID generated by the implementation. All subsequent calls to proc_DeleteUrl within the SQL transaction MUST pass the value returned by the first call.

@eventData: This parameter is reserved for internal use and MUST be the default value.

@acl: This parameter is reserved for internal use and MUST be the default value.

@IsCascadeDeleteOperation: This parameter is reserved for internal use and MUST be the default value.

@IsCascadeParent: This parameter is reserved for internal use and MUST be the default value.
@ChildDeleteTransactionId: This parameter is reserved for internal use and MUST be the default value.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_DeleteUrl stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The specified URL was not found.</td>
</tr>
<tr>
<td>5</td>
<td>The user is not authorized to make this change.</td>
</tr>
<tr>
<td>33</td>
<td>Cannot delete a folder containing checked-out or locked files.</td>
</tr>
<tr>
<td>36</td>
<td>The number of rows deleted by the operation exceeds the threshold.</td>
</tr>
<tr>
<td>50</td>
<td>Cannot delete a site (2) or an attachments folder.</td>
</tr>
<tr>
<td>51</td>
<td>Cannot delete a forms folder.</td>
</tr>
<tr>
<td>138</td>
<td>Cannot delete a URL containing lists.</td>
</tr>
<tr>
<td>161</td>
<td>Cannot delete a URL that contains sites.</td>
</tr>
<tr>
<td>206</td>
<td>The URL is too long.</td>
</tr>
<tr>
<td>1150</td>
<td>A concurrency violation or unknown error occurred.</td>
</tr>
<tr>
<td>1359</td>
<td>Internal error, or bad parameter specified (Attachments Flag specifies an attachment file, but the URL to be deleted is a file in a document library).</td>
</tr>
<tr>
<td>4335</td>
<td>Cannot delete a URL because of a lookup relationship referential integrity constraint.</td>
</tr>
<tr>
<td>8398</td>
<td>There was an error deleting a list. At least one list could not be deleted.</td>
</tr>
</tbody>
</table>

The proc_DeleteUrl stored procedure MUST return zero, one, or two result sets in the order shown.

3.1.5.13.1 Deleted Documents Result Set

The Deleted Documents Result Set contains information about the deleted documents. It MUST be returned on successful completion when @ListDeletedUrls is set to 1.

If the Deleted Documents Result Set is returned, it MUST contain one row for each document deleted.

If the document specified by @Url has a Document Store Type (section 2.2.4) of 0 (File), then the columns in the Deleted Documents Result Set MUST NOT be named; else they are named as follows:

| (Url) nvarchar(260), Type tinyint |

Url: The store-relative form URL of the deleted document.

Type: The Document Store Type of the deleted document.

3.1.5.13.2 Deleted Aliased Lists Result Set
The **Deleted Aliased Lists Result Set** contains information about all deleted lists that were configured with an e-mail address. Such lists provide implementation-specific e-mail integration features. The **Deleted Aliased Lists Result Set** MUST be returned only when `@ListDeletedAliases` is set to 1 and the document specified by `@Url` does not have a Document Store Type (section 2.2.2.4) of 0 (File).

If the **Deleted Aliased Lists Result Set** is returned, there MUST be one row returned for each deleted alias list.

```sql
{webSiteId} uniqueidentifier,  
 tp_webId uniqueidentifier,  
 tp_id uniqueidentifier;
```

*{webSiteId}*: The site collection identifier of the site collection containing the deleted list.

*tp_webId*: The site identifier of the *site (2)* containing the list.

*tp_id*: The list identifier of the deleted list.

### 3.1.5.13.3 Empty Deleted Aliased Lists Result Set

The **Empty Deleted Aliased Lists Result Set** MUST be returned only if `@ListDeletedAliases` is set to 1 and the document specified by `@Url` has a Document Store Type (section 2.2.2.4) of 0 (File). Deletion of a file does not cause the deletion of lists, and the **Empty Deleted Aliased Lists Result Set** is returned to indicate that the input parameter `@ListDeletedAliases` cannot be satisfied.

The **Empty Deleted Aliased Lists Result Set** has zero rows with a schema of two unnamed columns.

### 3.1.5.14 proc_DisableRbs

The **proc_DisableRbs** stored procedure is invoked to prepare the back-end database server to stop storing content in remote BLOB storage, and reclaim any back-end database server resources allocated with **proc_EnableRbs**.

```sql
PROCEDURE proc_DisableRbs (  
 @RequestGuid uniqueidentifier = null OUTPUT,  
);
```

*@RequestGuid*: The optional request identifier for the current request

**Return values:** The **proc_DisableRbs** stored procedure returns an integer return code, which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1060</td>
<td>The back-end database server does not support remote BLOB storage.</td>
</tr>
<tr>
<td>108</td>
<td>The back-end database server cannot disable remote BLOB storage because content is still stored in remote BLOB storage.</td>
</tr>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
</tbody>
</table>

The **proc_DisableRbs** stored procedure MUST NOT return any result sets.
### 3.1.5.15 proc_DirtyDependents

The **proc_DirtyDependents** stored procedure is invoked to mark all items that depend on a given document, configuration setting, navigation structure, or usage statistic as "dirty", so that subsequent action can be taken to update them as necessary. In this context, a "dependent" item is an item that requires an update to its metadata when another item is modified. **proc_DirtyDependents** follows the full dependency chain until all dependent items are marked.

```sql
PROCEDURE proc_DirtyDependents(
    @SiteId uniqueidentifier,
    @DepType tinyint,
    @DepDesc nvarchar(270) = NULL,
    @DepDescLike nvarchar(260) = NULL,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

- **@SiteId**: The site collection identifier for a site collection that contains the items with dependencies.

- **@DepType**: The dependency type. The following values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Document dependency. This updates items dependent on the specified document. The @DepDesc parameter is the store-relative form URL of the document that has changed.</td>
</tr>
<tr>
<td>3</td>
<td>Configuration dependency. This updates items dependent on changes to system configuration metadata, as specified in [MC-FPSEWM] section 2.2.2.3. The @DepDesc parameter is the metakey for the metadata that has changed.</td>
</tr>
<tr>
<td>4</td>
<td>Navigation dependency. This updates items dependent on changes to navigation structures. The @DepDesc parameter contains the Web-Navigation-URL, as specified in [MC-FPSEWM] section 2.2.2.34, for a navigation structure node.</td>
</tr>
<tr>
<td>7</td>
<td>Usage dependency. This updates items dependent on changes to site (2) usage statistics. The @DepDesc parameter is the store-relative form URL of the site (2).</td>
</tr>
</tbody>
</table>

- **@DepDesc**: This specifies the dependency description parameter, which varies according to the value of @DepType, as described in the preceding table.

- **@DepDescLike**: This parameter is unused and MUST be ignored.

- **@RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_DirtyDependents** stored procedure MUST return value 0.

The **proc_DirtyDependents** stored procedure returns no result sets.

### 3.1.5.16 proc_EnableRbs

The **proc_EnableRbs** stored procedure is invoked to prepare the back-end database server to store content in remote BLOB storage.

```sql
PROCEDURE proc_EnableRbs (  
    @RequestGuid uniqueidentifier = null OUTPUT,
);
```

- **@RequestGuid**: The optional request identifier for the current request.
Return values: The *proc_EnableRbs* stored procedure returns an integer return code, which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1060</td>
<td>The back-end database server does not support remote BLOB storage.</td>
</tr>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
</tbody>
</table>

The *proc_EnableRbs* stored procedure MUST NOT return any result sets.

### 3.1.5.17 proc_GetList

The *proc_GetList* stored procedure returns a list of the lists (1) in a site (2), along with their associated metadata.

```sql
PROCEDURE proc_GetList (  
    @SiteId                        uniqueidentifier,  
    @WebId                         uniqueidentifier,  
    @Collation                     nvarchar(48),  
    @BaseType                      int                =NULL,  
    @BaseType2                     int                =NULL,  
    @BaseType3                     int                =NULL,  
    @BaseType4                     int                =NULL,  
    @ServerTemplate                int                =NULL,  
    @FMobileDefaultViewUrl         bit                =NULL,  
    @FRootFolder                   bit                =NULL,  
    @ListFlags                     int                =NULL,  
    @FAclInfo                      int                =NULL,  
    @Scopes                        varbinary(max) =NULL,  
    @FRecycleBinInfo               bit                =NULL,  
    @UserId                        int                =NULL,  
    @FGP                           bit                =NULL,  
    @RequestGuid                   uniqueidentifier   =NULL OUTPUT  
);  
```

@SiteId: The site collection identifier of the site collection.

@WebId: The site identifier for the site (2) containing the requested lists (1).

@Collation: This parameter determines the sort order in which the lists are returned in the List Information Result Set (section 3.1.5.17.1) (based on the tp_Title column). This MUST be the Windows collation name of one of the valid Collation Order Enumeration values, with the case-insensitive and accent-sensitive flags set. For example, the collation order "Latin1_General" with the case-insensitive and accent-sensitive flags set, has a Windows collation name of "Latin1_General_CI_AS".

@BaseType: A parameter that restricts the returned lists by List Base Type (section 2.2.1.2.11). If this parameter is set one of the List Base Type values, the lists returned MUST be restricted to lists with this List Base Type and with the List Base Types specified by @BaseType2, @BaseType3, and @BaseType4, if any. If this parameter is set to "-1" or NULL, then the lists matching any List Base Type value will be returned, and parameters @BaseType2, @BaseType3, and @BaseType4 MUST be ignored by this procedure.

@BaseType2, @BaseType3, @BaseType4: Additional parameters used to specify additional List Base Types for lists returned, depending on the value of the @BaseType parameter. If any of these additional parameters are set to "-1" or NULL, then the additional parameter MUST be ignored.

@ServerTemplate: The identifier for the List Server Template (section 2.2.1.2.12) that defines the base structure of the lists to be returned. If this parameter is set to one of the List Server Template
values, then those lists returned MUST be restricted to those whose List Server Template value matches @ServerTemplate. If this parameter is set to "-1" or NULL, then lists matching any List Server Template will be returned.

@FMobileDefaultViewUrl: A bit specifying whether information about the mobile default view URL is requested. If this parameter is set to 1, then the information MUST be returned in the tp_MobileDefaultViewUrl column in the List Information Result Set. Otherwise, the tp_MobileDefaultViewUrl column MUST be returned with NULL values.

@FRootFolder: A bit specifying whether information about the root folder URL is requested. If this parameter is set to 1, then the information MUST be returned in the tp_RootFolder column in the List Information Result Set. Otherwise, the tp_RootFolder column MUST be returned with NULL values.

@ListFlags: A List Flags (section 2.2.2.5) value restricting the data returned by the List Information Result Set. If this parameter is not NULL, then the stored procedure MUST only return data in the List Information Result Set for lists with a tp.FLAGS value exactly matching the @ListFlags value. Otherwise, this parameter MUST be ignored.

@FAcclInfo: A flag specifying whether ACL information is requested. If this parameter is set to 1, then the ACL information MUST be returned in the tp_ACL column in the List Information Result Set. Otherwise, the tp_ACL column MUST be returned with NULL values.

@Scopes: A concatenation of one or more Scope Identifiers (section 2.2.1.1.8), represented as 16-byte binary strings with no delimiters, each specifying a scope identifier. If this parameter is not NULL, then this specifies that the stored procedure MUST only return data in the List Information Result Set for lists matching those scopes specified by this parameter. Otherwise, this parameter MUST be ignored.

@FRecycleBinInfo: A bit specifying whether to return information about the contents of the implementation-specific Recycle Bin. If this parameter is set to 1, then the Recycle Bin Information Result Set (section 3.1.5.17.2) MUST be returned. Otherwise, the NULL Result Set (section 3.1.5.17.3) MUST be returned.

@UserId: A User Identifier (section 2.2.1.1.13) that specifies the information to return in the Recycle Bin Information Result Set. If the @FRecycleBinInfo parameter is set to 1, then the stored procedure MUST only return data in the Recycle Bin Information Result Set. Otherwise, this parameter MUST be ignored.

@FGP: A bit restricting the data returned by the List Information Result Set. If this parameter is set to 1, then the stored procedure MUST only return data in the Recycle Bin Information Result Set for lists that are using per-list permissions. Otherwise, this parameter MUST be ignored.

@RequestGuid: The optional request identifier for the current request.

Return Values: The procEnumLists stored procedure returns an integer return code, which MUST be 0. This stored procedure MUST return two result sets in the order specified in the following sections.

### 3.1.5.17.1 List Information Result Set

The List Information Result Set returns information about lists (1) contained in the specified site (2). This result set MUST be returned and MUST contain zero or more rows, one for each list that matches the specified input parameters.
tp_DocTemplateUrl: Contains the store-relative URL of the document template associated with the list, if any, or NULL if none.

tp_DefaultViewUrl: Contains the URL in store-relative form default view of the list. This value MUST NOT be NULL.

tp_MobileDefaultViewUrl: When @FMobileDefaultViewUrl is set to "1", this contains the URL in store-relative form of the default view for mobile devices, if any, or NULL if none has been set. Otherwise, this MUST be NULL.

tp_Id: Contains the List Identifier (section 2.2.1.1.5) for the list.

tp_Title: Contains the display name (a short user-provided text description) to identify the list.

tp_Description: Contains user-provided text describing the list.

tp_ImageUrl: Contains the URL in store-relative form holding an image associated with the list.

tp_Name: Contains a string representation of the tp_Id GUID delimited by braces ({}).

tp_BaseType: Contains the List Base Type (section 2.2.1.2.11) value from which the list is derived.

tp_FeatureId: Contains a Feature Identifier (section 2.2.1.1.4) for a feature associated with the list.
**tp_ServerTemplate**: Contains the value of the of the list template that defines the base structure of the list. This value can either be in the List Server Template (section 2.2.1.2.12) enumeration or can be a custom value as defined in the disk-based template of the list. A custom list template value MUST be unique and MUST be greater than 10000.

**tp_Created**: Contains the time in Coordinated Universal Time (UTC), specifying when the list was created.

**tp_Modified**: Contains the time in UTC, specifying when the list was last modified.

**tp_LastDeleted**: Contains a time stamp, in UTC format, specifying when a list item was last deleted from the list. If no list item has been deleted, contains the value of tp_Created.

**tp_Version**: Contains an implementation-specific, internal version counter used in tracking modifications to the list’s settings.

**tp_Direction**: Contains a value specifying the direction of text flow for front-end Web server elements. Valid values are in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No explicit direction is specified.</td>
</tr>
<tr>
<td>1</td>
<td>Text flow is left to right.</td>
</tr>
<tr>
<td>2</td>
<td>Text flow is right to left.</td>
</tr>
</tbody>
</table>

**tp_ThumbnailSize**: The width, in pixels, specified for use when creating thumbnail images of list items within this list. This value MUST be NULL for lists that do not have a List Server Template value of 109. Thumbnails images are generated by the front-end Web server for documents and they are implementation-specific capabilities.

**tp_WebImageWidth**: The width, in pixels, specified for use when creating images of list items within this list. This value MUST be NULL for lists that do not have a List Server Template value of 109.

**tp_WebImageHeight**: The height, in pixels, specified for use when creating images of list items within this list. This value MUST be NULL for lists that do not have a List Server Template value of 109.

**tp.Flags**: Contains a List Flags (section 2.2.2.5) value describing list properties.

**tp.ItemCount**: Contains a count of list items in the list.

**tp_AnonymousPermMask**: Contains the WSS Rights Mask (section 2.2.2.15) that applies to an anonymous user for the list.

**tp_RootFolder**: When the @FRootFolder parameter is set to "1", this contains the URL in store-relative form of the root folder of the list. Otherwise, this MUST be NULL.

**tp_ReadSecurity**: Contains a value identifying the security policy for read-only mode for list items. If this value is set to 1, then users with read-only mode permissions can read all list items. Otherwise, users with read-only mode permissions can read only those list items they create.

**tp_WriteSecurity**: Contains a value specifying the security policy in use for write access to list items. Valid values are in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Users with write permissions have write access to all list items.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>2</td>
<td>Users with write permissions have write access to those list items they create.</td>
</tr>
<tr>
<td>4</td>
<td>Users have no write access to any list items.</td>
</tr>
</tbody>
</table>

**tp_Author**: Contains the User Identifier (section 2.2.1.13) of the user who created the list.

**tp_EventSinkAssembly**: Contains the assembly name of the event sink handler if an event sink handler is registered for the list or NULL if none exists.

**tp_EventSinkClass**: Contains the assembly class name of the event sink handler if an event sink handler is registered for the list or NULL if none exists.

**tp_EventSinkData**: Contains string data specific to the implementation of the event sink associated with this list, or NULL if none exists.

**tp_EmailAlias**: Contains the e-mail address of the list. This alias is used to allow files to be sent directly to the list through an implementation-specific e-mail handling feature. Can be NULL if the list is not an e-mail enabled list.

**tp_WebFullUrl**: Contains the store-relative URL of the site (2) that contains the list.

**tp_WebId**: Contains the Site Identifier (section 2.2.1.11) of the site (2) that contains the list.

**tp_SendToLocation**: Contains an implementation-specific string of the URL used to copy list items to alternative locations. This parameter can be NULL.

**tp_ScopeId**: Contains the scope identifier for the list.

**{tp_MaxMajorVersionCount}**: Contains the number of major versions that will be retained for this document if versioning is enabled.

**{tp_MaxMajorVersionCount}**: Contains the number of major versions that will have their associated minor versions retained for this document if versioning is enabled.

**tp_DefaultWorkflowId**: Contains the Workflow Identifier (section 2.2.1.16) of the default workflow associated with the list or NULL if none exists.

**tp_HasInternalFGP**: This flag is set to 1 if there have ever been list items that have had per-list permissions applied.

**tp_NoThrottleListOperations**: If this list is exempt from resource throttling operations, then this value MUST be 1. Otherwise, it MUST be 0.

**HasRelatedLists**: This value MUST be NULL.

**Followable**: If this list is followable, then this value MUST be 1. Otherwise, it MUST be 0.

**tp_ValidationFormula**: This value MUST be NULL.

**tp_ValidationMessage**: This value MUST be NULL.

**TitleResource**: This value MUST be NULL.

**DescriptionResource**: This value MUST be NULL.

**tp_ACL**: When the @FaclInfo parameter is set to 1, this MUST contain a binary array of the ACL for this list if one is defined; otherwise, it MUST be NULL.

### 3.1.5.17.2 Recycle Bin Information Result Set
The Recycle Bin Information Result Set returns information about list items from the specified list that the specified user marked as deleted in the Recycle Bin.

If the @FRecycleBinInfo parameter is 1, the Recycle Bin Information Result Set MUST be returned and MUST contain a single row with two columns specified as follows.

```
{RecycleBinCount} int,
{RecycleBinSize} int;
```

{RecycleBinCount}: Contains a count of the number of items in the Recycle Bin matching the specified user and list.

{RecycleBinSize}: Contains the total size, in bytes, of items in the Recycle Bin matching the specified user and list.

3.1.5.17.3 NULL Result Set

The NULL Result Set is an empty placeholder returned when Recycle Bin information is not requested. The NULL Result Set MUST only be returned if the @FRecycleBinInfo parameter is not 1 and MUST contain no rows, in a schema consisting of a single, unnamed NULL column.

3.1.5.18 proc_ReadStream

The proc_ReadStream stored procedure is called to read a specific range of data from a document's stream binary piece.

```
PROCEDURE proc_ReadStream ( 
    @SiteId               uniqueidentifier,
    @ParentId             uniqueidentifier,
    @DocId                uniqueidentifier,
    @Level                tinyint,
    @Partition            tinyint,
    @BSN                  bigint,
    @Offset               int,
    @Length               int
);
```

@SiteId: The site collection identifier for a site collection containing the document.

@ParentId: The document identifier of the document's parent container.

@DocId: The document identifier of the document to read from.

@Level: The publishing level of the document.

@Partition: The stream partition which contains the stream binary piece to read.

@BSN: The BLOB sequence number of the stream binary piece to read.

@Offset: The offset, in bytes, into the stream binary piece to begin reading.

@Length: The number of bytes to read from the stream binary piece.

Return Values: This stored procedure returns an integer return code which MUST be listed in the following table.
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>30</td>
<td>There was an IO error reading the data.</td>
</tr>
</tbody>
</table>

Result Sets: The proc_ReadStream stored procedure MUST return the **Document Stream Chunk Result Set** (section 3.1.5.18.1).

### 3.1.5.18.1 Document Stream Chunk Result Set

The **Document Stream Chunk Result Set** contains a range of bytes read from a document's stream binary piece.

```
{Content}                         varbinary(max);
```

**{Content}**: The data read from the document's stream binary piece.

### 3.1.5.19 proc_FetchDocForHttpGet

The **proc_FetchDocForHttpGet** stored procedure is called to fetch a **document stream** and provide information necessary to render the document on a **front-end Web server**.

```sql
PROCEDURE proc_FetchDocForHttpGet(
    @DocSiteId                     uniqueidentifier,
    @DocDirName                    nvarchar(256),
    @DocLeafName                   nvarchar(128),
    @LooksLikeAttachmentFile       bit,
    @IfModifiedSince               datetime,
    @FetchType                     int,
    @ValidationType                int,
    @ClientVersion                 int,
    @ClientId                      uniqueidentifier,
    @PageView                      tinyint,
    @FetchBuildDependencySet       bit,
    @SystemID                      varbinary(512),
    @AppPrincipalName              nvarchar(256),
    @IsHostHeaderAppPrincipalName  bit,
    @CurrentVirusVendorID          int,
    @PrefetchListScope             bit,
    @ChunkSize                     int,
    @DGCacheVersion                bigint,
    @MaxCheckinLevel               tinyint,
    @HonorLevel                    bit,
    @CurrentFolderUrl              nvarchar(260),
    @ThresholdRowCount             int,
    @StreamPartition               tinyint,
    @Level                         tinyint            OUTPUT,
    @FetchStreamIfNeeded           bit = 1,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

**@DocSiteId**: The **site collection identifier** for a **site collection** containing the document.

**@DocDirName**: The **directory name** of the requested document.

**@DocLeafName**: The **leaf name** of the requested document.

**@LooksLikeAttachmentFile**: **Boolean** flag that specifies if the document is an attachment to a **list item** or an attachment folder.
@IfModifiedSince: Specifies the datetime in UTC of the last modified time stamp of the document. If @ValidationType is "2" or "3", the document MUST only be fetched if this value is earlier than the current last modified time stamp value in the back-end database server.

@FetchType: Specifies the type of request. If this parameter is "0", information to satisfy an HTTP GET request MUST be returned, including the stream of the document. If this parameter is "1", information needed only to satisfy an HTTP HEAD request MUST be returned, which does not include the stream of the document. All nonzero values MUST also be treated as "1".

@ValidationType: This parameter MUST have a value of "0", "1", "2", or "3". If this parameter is "0", @ClientVersion, @ClientId, and @IfModifiedSince MUST be ignored and the document MUST be fetched.

@ClientVersion: Specifies the internal version number of the document to fetch. If @ValidationType is "1" or "3", the document MUST only be fetched if this value is equal to the current internal version number of the document on the back-end database server.

@ClientId: Specifies the document identifier of the document to fetch. If @ValidationType is "1" or "3", the document MUST only be fetched if this value is equal to the current document identifier of the document on the back-end database server.

@PageView: Non-NULL values indicate that the document is a view Web page or an implementation-specific Web page that renders an item or items in a list (1) or document library. This also indicates that information is needed to enable front-end Web server rendering based on metadata about the view page, the item or items being rendered, the user's browsing context, the overall site's navigation scheme, and the user's security privileges. Information requested includes site metadata, containing list (1) metadata, users and permission levels metadata, Web Parts, and related list (1) metadata. In addition, a value of one indicates that the metadata is security trimmed to the user specified by @SystemID. The value zero or other non-NULL values indicate that information is requested as seen by all users. A NULL value indicates that the document is not a view Web page.

@FetchBuildDependencySet: If this parameter is "1" and the document being fetched has a publishing level of published, the Document Build Dependency Set Result Set (section 3.1.5.19.12) and Document Build Dependency Metadata Result Set (section 3.1.5.19.13) MUST be returned. Otherwise, the result sets MUST NOT be returned.

@SystemID: The security identifier (SID), as specified in [MS-DTYP] section 2.4.2, of the current user, or NULL to indicate an anonymous user.

@AppPrincipalName: The app principal identifier or the app web domain identifier associated with the current user. If the current user is not associated with an app principal, this parameter MUST be NULL.

@IsHostHeaderAppPrincipalName: If this parameter is set to one, the @AppPrincipalName parameter MUST be the app principal identifier, else it MUST be the app web domain identifier.

@CurrentVirusVendorID: Specifies an identifier for an anti-virus scanner. If this value is not NULL and does not match the current anti-virus scanner registered for the document, the document MUST NOT be fetched.

@PrefetchListScope: If this parameter is set to "1", the metadata of the list (1) scope that contains the document MUST be returned.

@ChunkSize: Specifies the maximum size, in bytes, of the document content to be returned in the {Content} column in the Document Information (Get) Stream Result Set (section 3.1.5.19.9). For an uncustomized document, this MUST be NULL. Otherwise, if the data in the {Content} column is larger than the value specified in the @ChunkSize parameter, only the first @ChunkSize bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.
@DGCacheVersion: Specifies an internal version number for the domain group map cache of the site collection that contains this document being request. If this parameter is "-2", the Domain Group Cache BEDS Update Result Set (section 3.1.5.19.3) and the Domain Group Cache WFE Result Set (section 3.1.5.19.4) MUST NOT be returned. Otherwise, the Domain Group Cache BEDS Update Result Set MUST be returned, and the Domain Group Cache WFE Update Result MUST only be returned if @DCCacheVersion is less than the current internal version number of the domain group map cache.

@MaxCheckinLevel: Specifies the publishing level of the document to fetch. If @HonorLevel is equal to "0", this parameter MUST be ignored.

@HonorLevel: If this parameter is not equal to "0", the document with publishing level specified in @MaxCheckInLevel MUST be fetched. Otherwise, the document with the maximum publishing level available to the current user MUST be fetched.

@CurrentFolderUrl: The URL of the folder that the user was browsing before the specified document was requested. This is used to preserve the browsing context when the fetch request is redirected to a view page. When fetching an item in a list (1) or a file in a document library, an attachment to an item, a folder, or a list view, the specified document will be a view page, and @CurrentFolderUrl will be used to render the browsing context.

@ThresholdRowCount: Specifies the maximum number of security scopes to be fetched. If @PrefetchListScope is not equal to "1", this parameter MUST be ignored.

@StreamPartition: Specifies the stream partition of the document to fetch.

@Level: An output parameter specifying the publishing level of the document to be fetched.

@FetchStreamIfNeeded: Specifies if the stream of the document is fetched.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_FetchDocForHttpGet stored procedure returns an integer return code, which MUST be one of those listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>2</td>
<td>The document specified by @DocSiteId, @DocDirName, @DocLeafName (and @MaxCheckinLevel, if provided) was not found, or the document is an attachment folder; or the document is a folder, but no welcome page, home page, view Web page, or other redirect Web page was found for it.</td>
</tr>
<tr>
<td>18</td>
<td>Successful execution; based on values of @ValidationType, @ClientId, @ClientVersion, and @IfModifiedSince, the document was not fetched.</td>
</tr>
<tr>
<td>146</td>
<td>The document does not exist, but a welcome page was found, as specified by the Welcome Page Redirect Information Result Set.</td>
</tr>
<tr>
<td>1168</td>
<td>The site collection specified by @DocSiteId was not found.</td>
</tr>
<tr>
<td>1271</td>
<td>The site (2) is read locked.</td>
</tr>
</tbody>
</table>

Result Sets: The proc_FetchDocForHttpGet stored procedure MUST return zero or more result sets upon successful execution. If the document is not found, zero result sets MUST be returned. The following result sets MUST be returned only if the document has a Document Store Type (section 2.2.2.4) of zero (File):

- **Site Collection Audit Mask Result Set** (section 3.1.5.19.10)
- **List Audit Mask Result Set** (section 3.1.5.19.11)
Other proc_FetchDocForHttpGet result sets return conditionally, as described in each result set section, in the following sections.

### 3.1.5.19.1 HTTP Document Metadata Result Set

The **HTTP Document Metadata Result Set** returns the core document metadata, including the effective security permission and anonymous permission mask. This result set MUST be returned if the specified document exists.

```sql
[Size]                        int,
[DocFlags]                    int,
[FullUrl]                     nvarchar(260),
[WebId]                       uniqueidentifier,
[FirstUniqueWebId]            uniqueidentifier,
[SecurityProvider]           uniqueidentifier,
[Dirty]                      bit,
[TimeLastWritten]            datetime,
[CharSet]                    int,
[Version]                    int,
[DocId]                      uniqueidentifier,
[LeafName]                   nvarchar(128),
```
InDocLibrary bit,
IsAttachment bit,
NeedManageListRight int,
{SiteFlags} int,
Acl varbinary(max),
AnonymousPermMask bigint,
{ListIdForPermissionCheck} uniqueidentifier,
{PermCheckedAgainstUniqueList} int,
DraftOwnerId int,
ListFlags bigint,
Level tinyint,
{IsCurrentVersion} bit,
{Type} tinyint,
{VirusVendorID} int,
{VirusStatus} int,
{VirusInfo} nvarchar(255),
{VirusInfoEx}, varbinary(max),
{ContentModifiedSince} bit,
{ProgId} nvarchar(255),
{DoclibRowId} int,
{Language} int,
{DirName} nvarchar(255),
{UIVersion} int,
{RbsCollectionId} int,
{NextBSN} bigint,
{StreamSchema} byte,
{InternalVersion} int,
{WebFlags} int,
{AppWebDomainId} varchar(8),
{SiteAppHostHeader} nvarchar(55),
{DocScopeId} uniqueidentifier,
{DenyPermMask} tPermMask

{Size}: The size, in bytes, of the document stream.

{DocFlags}: A Doc Flags value (section 2.2.2.3) describing the document.

{FullUrl}: The complete store-relative form URL for the requested document.

{WebId}: The site identifier of the site (2) containing the document.

{FirstUniqueWebId}: The site identifier of the site (2) whose security permissions are the effective security permission for the site (2) containing the specified document.

{SecurityProvider}: COM CLSID of the security provider for this site (2).

{Dirty}: A bit that specifies this document MUST have implementation-specific processing performed before its stream can be returned. If this document does not require processing, this value MUST be zero. If this document does not have a stream, this value MUST be NULL.

{TimeLastWritten}: The datetime, in UTC, when the document stream was last modified.

{CharSet}: A character set associated with the document. This value MUST be NULL or a valid Windows code page identifier.

{Version}: The internal version number of the document being returned.

{DocId}: The document identifier of the requested document.

{LeafName}: The leaf name of the requested document.
InDocLibrary: If the document is in a document library, this value MUST be "1".

IsAttachment: If the document is an attachment, this value MUST be "1".

NeedManageListRight: If the user is required to have the Manage List right to read the document, this value MUST be "1".

{SiteFlags}: A Site Collection Flags value (section 2.2.9) describing the configuration of the site collection containing the document.

Acl: The WSS ACL Format (section 2.2.6) permissions for the document.

AnonymousPermMask: A WSS Rights Mask (section 2.2.15) that indicates the rights granted to an anonymous user on this document.

{ListIdForPermissionCheck}: The GUID for the list (1) containing the document. This value MUST be NULL if this document is not in a list (1).

{PermCheckedAgainstUniqueList}: MUST be zero.

DraftOwnerId: The identifier for the user who published the document as a draft version. If the document is not a draft version, this value MUST be NULL.

ListFlags: A List Flags value (section 2.2.5) describing the list (1) that contains the document. If the document is not in a list (1), the value MUST be zero.

Level: The publishing level value of this document.

{IsCurrentVersion}: If the document being returned is the current version, as defined by the implementation, this value MUST be "1", else this value MUST be "0".

{Type}: The Document Store Type (section 2.2.4) of this document.

{VirusVendorID}: The identifier of the anti-virus vendor that processed this document. This value MUST be NULL if the document has not been processed by an anti-virus scanner.

{VirusStatus}: The Virus Status (section 2.2.1.2.17) of the document. This value MUST be NULL if the requested document does not exist or if it has not been processed by an anti-virus scanner.

{VirusInfo}: An anti-virus scanner specific message returned by the anti-virus scanner when it last processed the document.

{VirusInfoEx}: A binary value that specifies extensible virus information for this document, which is opaque to the back-end server.

{ContentModifiedSince}: A bit indicating whether the document has been modified, depending on the validation type in use. This MUST be set to one if any of the following are true:

- the document is a dynamic document type.
- the document requires a dependency update.
- validation type is "None" (0).
- validation type is "E-tag" (1) and the value of @ClientVersion disagrees with the document version in the store or the value of @ClientId disagrees with the document identifier in the store.
- validation type is "Last modified" (2) and the last modification date of the document in the store is more recent than specified in @IfModifiedSince.

In all other cases, ContentModifiedSince MUST be set to zero.
{ProgId}: Specifies an implementation-specific preferred application to open the document.

{DoclibRowId}: The identifier of the list item which represents this document if it belongs in a list (1). If the requested document is not contained in a list (1), this value MUST be NULL.

{Language}: The LCID of the locale of the site (2).

{DirName}: The directory name of the requested document.

{UIVersion}: The displayed version number of the document being returned.

{ContentVersion}: The version number of the document stream being returned.

{RbsCollectionId}: The identifier for the remote BLOB storage collection for the site collection, or zero if remote BLOB storage is not configured for this database.

{NextBSN}: The current BLOB sequence number of the document being returned.

{StreamSchema}: The current stream schema of the document being returned.

{InternalVersion}: The current internal version number of the document being returned.

{WebFlags}: A Site Property Flags value (section 2.2.2.11) describing the configuration of the site (2) containing the document.

{AppWebDomainId}: The app web domain identifier of the site (2) containing the document being returned.

{SiteAppName}: The app host header name of the site collection containing the document being returned.

{DocScopeId}: The GUID identifying the security scope of the document. This value MUST NOT be NULL.

{DenyPermMask}: A WSS Rights Mask (section 2.2.2.15) specifying the rights to deny to the current user.

3.1.5.19.2  Domain Group Cache Versions Result Set

The Domain Group Cache Versions Result Set contains information about the version numbers associated with the Domain Group Map Caches on the front-end Web server and on the back-end database server for the specified site collection.

The Domain Group Cache Versions Result Set MUST be returned if the specified document exists and MUST contain one row of version number data. If the specified @DGCacheVersion value is -2, then all columns returned MUST have the value -2, indicating that the value MUST NOT be used for comparison. The Domain Group Cache Versions Result Set is defined in the Common Result Sets Domain Group Cache Versions Result Set (section 2.2.4.4).

3.1.5.19.3  Domain Group Cache BEDS Update Result Set

The Domain Group Cache BEDS Update Result Set contains information to be used in recomputing the Domain Group Map Cache.

The Domain Group Cache BEDS Update Result Set MUST be returned if @DGCacheVersion is not -2 ('Skip') and the real Domain Group Map Cache version is more recent than the cached version on the back-end database server; that is, if the value of RealVersion is greater than the value of CachedVersion in the Domain Group Cache Versions Result Set (section 2.2.4.4).
If the Domain Group Cache BEDS Update Result Set is returned, it indicates that the copy of the Domain Group Map Cache on the back-end database server is out of date and MUST be recomputed to ensure that proper security checks can be made.

When returned, the Domain Group Cache BEDS Update Result Set MUST have a single row. The Domain Group Cache BEDS Update Result Set is defined in the Common Result Sets Domain Group Cache BEDS Update Result Set (section 2.2.4.3).

### 3.1.5.19.4 Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the domain group map cache.

The Domain Group Cache WFE Update Result Set MUST be returned only if @DGCacheVersion is not -2 ("Skip") and the cached version on the back-end database server is up-to-date; that is, if the value of RealVersion is not greater than the value of CachedVersion in the Domain Group Cache BEDS Update Result Set (section 2.2.4.3).

The Domain Group Cache WFE Update Result Set is defined in the Common Result Sets Domain Group Cache WFE Update Result Set (section 2.2.4.5).

### 3.1.5.19.5 User Information Result Set

The User Information Result Set returns information about the user specified in @UserId.

```sql
    tp_Id int,
    tp_SiteAdmin bit,
    tp_IsActive bit,
    tp_Login nvarchar(255),
    tp_Email nvarchar(255),
    tp_Title nvarchar(255),
    tp_Notes nvarchar(1023),
    tp_ExternalTokenLastUpdated datetime,
    tp_Token varbinary(max),
    UserId int,
    SiteSecurityVersion bigint;
```

**tp_Id**: The user identifier.

**tp_SiteAdmin**: Indicates whether the specified user is a site collection administrator on the site collection specified by @DocSiteId.

**tp_IsActive**: MUST be set to "1" if the specified user is an active user in the site collection specified by @DocSiteId.

**tp_Login**: The login name of the specified user.

**tp_Email**: The e-mail address of the specified user.

**tp_Title**: The display name of the specified user.

**tp_Notes**: Notes about the specified user.

**tp_ExternalTokenLastUpdated**: The date and time, in UTC format, when the External Group Token for the specified user was last updated.

**tp_Token**: A WSS User Token value specifying the permission level membership of the specified user.
tp_Flags: A WSS User Flags value for the specified user.

UserId: The site membership identifier of the specified user. This parameter can be NULL if the user has not been added as a member to the site (2) whose permissions are in effect on the document.

SiteSecurityVersion: The current security information version of the site collection containing this document.

3.1.5.19.6 Welcome Page Redirect Information Result Set

The Welcome Page Redirect Information Result Set returns whether or not the document is a site (2) or a folder with a configured welcome page, or the welcome page itself. The Welcome Page Redirect Information Result Set contains information about the welcome page.

{RedirectType}: The Redirect Type of the URL. MUST be 0, indicating a welcome page redirect URL.

{RedirectUrl}: The URL of the welcome page.

WelcomePageParameters: This MUST contain any URL parameters configured for the welcome page. This value can contain a query string starting with "?" or a hash parameter starting with "#".

{ContentTypeId}: This value MUST be NULL.

3.1.5.19.7 Non-Welcome Page Redirect Information Result Set

The Non-Welcome Page Redirect Information Result Set returns if the document is a site (2) or a folder, where a welcome page is not configured. Depending on the specified document URL, the redirect can be to a home page, a list view Web page, or to a provisioning page URL. The Non-Welcome Page Redirect Information Result Set contains information about the redirect URL. The Non-Welcome Page Redirect Information Result Set MUST return a single row.

{RedirectType}: The Redirect Type (section 2.2.1.2.15) of the URL. This parameter MUST NOT be 0. For all other valid values, see the Redirect Type section.

{RedirectUrl}: The full redirect URL.

{WelcomePageParameters}: MUST be NULL.

{ContentTypeId}: The document's content type identifier.

3.1.5.19.8 Document Information (Get) Metadata Result Set

This result set contains the metadata for the document's binary stream. This row set MUST be returned only if @FetchType is not set to 1, otherwise, the row set MUST NOT be returned.

This result set MUST return zero or one rows. If the document is modified, then a single row MUST be returned. A document is considered modified subject to the semantics indicated by the input.
parameter @ValidationType, or if its Virus Vendor ID has been updated since the client last retrieved it. When the document is modified, the return code 18 MUST be returned upon successful completion of proc_FetchDocForHttpGet. Otherwise, zero rows MUST be returned.

The value returned in the {HistVersion} column MUST be 0.

This result set is defined as **Document Content Metadata Result Set** (section 2.2.4.6) in the Common Result Sets (section 2.2.4).

### 3.1.5.19.9 Document Information (Get) Stream Result Set

This result set contains a row for each of the document's **stream binary pieces**. This row set MUST be returned only if @FetchType is not set to 1.

This result set MUST return zero or more rows. If @FetchStreamIfNeeded is not set to 1 or the document requires dependency update processing or the document is an uncustomized document, this row set MUST contain 0 rows.

This result set is defined as **Document Content Stream Result Set** (section 2.2.4.7) in the Common Result Sets (section 2.2.4).

### 3.1.5.19.10 Site Collection Audit Mask Result Set

The **Site Collection Audit Mask Result Set** contains the information about the **Audit Flags** associated with the site collection containing the specified document.

The **Site Collection Audit Mask Result Set** is defined in the Common Result Sets **Site Audit Mask Result Set** (section 2.2.4.22).

### 3.1.5.19.11 List Audit Mask Result Set

The **List Audit Mask Result Set** contains information about the **Audit Flags** associated with the list containing the document.

The **List Audit Mask Result Set** MUST return one row if the document is contained in a list; otherwise, zero rows MUST be returned.

```sql
    tp_Id                          uniqueidentifier,
    tp_AuditFlags                  int,
    tp_InheritAuditFlags           int,
    {GlobalAuditMask}              int,
    {URL}                          nvarchar(516)
```

**tp_Id**: The List Identifier of the list containing the document.

**tp_AuditFlags**: An **Audit Flags** value determining the operations to be tracked on the list.

**tp_InheritAuditFlags**: An **Audit Flags** value determining the operations to be tracked on the document as inherited from the document's container.

**{GlobalAuditMask}**: An **Audit Flags** value determining the operations to be tracked across the site collection that contains the document.

**{URL}**: The URL of the list containing the document.

### 3.1.5.19.12 Document Build Dependency Set Result Set
The **Document Build Dependency Set Result Set** returns a build dependency set for a published document. The **Document Build Dependency Set Result Set** MUST return only when @FetchBuildDependencySet is set to 1 and when the document is published.

```sql
BuildDependencySet varbinary(max);
```

**BuildDependencySet**: A binary array holding implementation-specific information about dependent documents. NULL indicates that there is no information about dependencies. A **BuildDependencySet** of size 0 indicates a document with no dependencies.

### 3.1.5.19.13 Document Build Dependency Metadata Result Set

The **Document Build Dependency Metadata Result Set** contains information about each document in the specified document's build dependency set. This result set MUST be returned only if the @FetchBuildDependencySet is set to one and the binary array returned in the **Document Build Dependency Set Result Set** (section 3.1.5.19.12) is not NULL and is not of zero-length.

- **DirName**: The directory name of the document.
- **LeafName**: The leaf name of the document.
- **WebsFullUrl**: The store-relative form URL of the document.
- **FirstUniqueAncestorWebId**: The site identifier of the closest site (2) in this site's ancestor chain that does not inherit security settings from its parent site.
- **SecurityProvider**: COM CLSID of the external security provider for this site (2). This MUST be NULL for sites (2) using the native security implementation.
- **Acl**: The binary serialization of the WSS ACL Format (section 2.2.3.6) ACL for this site (2).
- **AnonymousPermMask**: A WSS Rights Mask (section 2.2.2.15) that indicates the rights granted to a user who is anonymous, or has no specific rights, to the document.
- **WebId**: The site identifier for a site (2) containing the document.
- **ListId**: The list identifier for the list (1) containing the document.
**UniqueList**: This value MUST be zero.

**InDocLibrary**: The value MUST be set to one if the document is contained within a document library. Otherwise, MUST be set to zero.

**DraftOwnerId**: The user identifier for the owner of the last checked-in draft version of the document.

**ListFlags**: A List Flags value (section 2.2.2.5) describing the list (1) that contains this document. This value MUST be NULL if the document is not stored in a list (1).

**ProgId**: Designates a preferred application to open this document. This value MUST be NULL if the document does not exist or the parser did not specify a Progid when the document was saved.

**SetupPathVersion**: For an uncustomized document, this governs the setup path location to which the SetupPath fragment is relative. This value MUST be NULL if the requested document does not exist and is undefined for documents that were never uncustomized. The following are values are valid.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).</td>
</tr>
<tr>
<td>3</td>
<td>The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).</td>
</tr>
<tr>
<td>4</td>
<td>Relative to the install location of Microsoft SharePoint Foundation 2010 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).</td>
</tr>
<tr>
<td>15</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Foundation 2013 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\15).</td>
</tr>
<tr>
<td>16</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Server 2016 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\16).</td>
</tr>
</tbody>
</table>

**SetupPath**: For a document that is now or once was uncustomized, this contains the setup path fragment relative to the base setup path described earlier by the SetupPathVersion value, where the content stream of this document can be found. This value MUST be NULL in the case of a document that does not exist or a document that was never uncustomized.

**TimeLastWritten**: A time stamp, in UTC format, specifying when any changes were made to the document stream. Does not reflect changes made strictly to the MetaInfo or other item properties.

**MasterUrl**: The URL for the master page registered on the site (2) for use in pages of the site (2) when rendered on the front-end Web server.

**CustomMasterUrl**: The URL for an alternate master page registered on the site (2) for use in pages of the site (2) rendered on the front-end Web server.

**Version**: A counter incremented any time a change is made to this document.

**Id**: The document identifier.

**BuildDependencySet**: A varbinary(max) value specifying further document dependencies.

### 3.1.5.19.14 Site Metadata Result Set
The **Site Metadata Result Set** contains metadata for the site (2) containing the specified document. The **Site Metadata Result Set** MUST return only if the input parameter @PageView is not NULL, as part of a series of result sets describing view Web page document metadata.

The **Site Metadata Result Set** is defined in the Common Result Sets **Site Metadata Result Set** (section 2.2.4.24).

### 3.1.5.19.15 Event Receivers Result Set

The **Event Receivers Result Set** contains information about the event receivers defined for the site (2) containing the specified document.

The Event Receivers Result Set is part of a series of result sets describing view Web page-related metadata. It MUST return only if the input parameter @PageView is not NULL, and it will not return if proc_FetchDocForHttpGet returns a return code of 1271, indicating that the site collection was locked.

The **Event Receivers Result Set** MUST contain one row per event receiver registered for the site (2). The **Event Receivers Result Set** is defined in the Common Result Sets **Event Receivers Result Set** (section 2.2.4.11).

### 3.1.5.19.16 Web Event Receiver Result Set

The **Web Event Receivers Result Set** contains information about the event receivers defined for the site (2) containing the specified document.

The **Web Event Receivers Result Set** is part of a series of result sets describing view Web page-related metadata. It MUST return only if the input parameter @PageView is not NULL, and it will not return if proc_FetchDocForHttpGet returns a return code of 1271, indicating that the site collection was locked.

The **Event Receivers Result Set** MUST contain one row per event receiver registered for the site (2). The Event Receiver Result Set is defined in the Common Result Sets **Event Receivers Result Set** (section 2.2.4.11).

### 3.1.5.19.17 Site Features List Result Set

The **Site Features List Result Set** returns information about available features. If the **Site Features List Result Set** returns, it MUST return twice: first, for site collection features, and then for site features, for the site (2) and site collection that contain the specified document.

The **Site Features List Result Set** MUST return only if @PageView is not NULL, and it will not return if proc_FetchDocForHttpGet returns a code of 1271, indicating that the site collection was locked.

The **Site Features List Result Set** is defined in the Common Result Sets **Site Feature List Result Set** (section 2.2.4.23).

### 3.1.5.19.18 WebParts Metadata, Personalized Result Set

The **Web Parts Metadata, Personalized Result Set** contains the core metadata about the Web Parts appearing on the specified document, personalized for the user specified in @UserId.

The **Web Parts Metadata, Personalized Result Set** is part of a series of result sets describing view Web page-related metadata. It MUST return only if the input @PageView is not 0.

The **Web Parts Metadata, Personalized Result Set** will not return if proc_FetchDocForHttpGet returns a code of 1271, indicating that the site collection was locked.

The **Web Parts Metadata, Personalized Result Set** MUST contain one row per Web Part.
tp_Id: The Web Part Identifier of the Web Part. This value MUST NOT be NULL.

tp_ZoneId: The name of a Web Part zone. This value can be NULL.

tp_WebPartTypeId: A 16-byte value uniquely identifying the type of the Web Part. MUST NOT be NULL.

tp_Assembly: The assembly name of the implementation of the Web Part.

tp_Class: The fully qualified class name of the implementation of the Web Part.

SolutionId: The solution identifier of the solution. MUST be NULL if there is no solution associated with the Web Part.

Hash: The implementation-specific hash of the content of the sandboxed solution. This value MUST be NULL if there is no solution associated with the Web Part.

ValidatorsHash: The implementation-specific hash of the sandboxed solution validators that validated the sandboxed solution. This value MUST be NULL if there is no solution associated with the Web Part.

ValidationErrorUrl: The URL to render if the sandboxed solution validator failed validation. This value MUST be NULL if there is no solution associated with the Web Part.

ValidationErrorMessage: The error message string to render if the sandboxed solution validator failed validation. This value MUST be NULL if there is no solution associated with the Web Part.

{tp_IsIncluded}: If this bit flag is set to 1, it indicates that the Web Part is visible. Otherwise, it MUST be 0.

{tp_FrameState}: A value that indicates the frame state of the Web Part. This value MUST be one of the following.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal. The Web Part is displayed in its normal state, with title, content, and placement within the page.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>Minimized. The Web Part is collapsed.</td>
</tr>
</tbody>
</table>

**tp** _AllUsersProperties_: Web Part Properties specified for all users. This value can be NULL.

**{tp_PerUsersProperties}:** Web Part Properties specified for per-user basis. This value can be NULL.

**{tp_PartOrder}:** Ordinal number that indicates the location of the Web Part in relation to other Web Parts in the same zone. This value can be NULL.

**{tp_Flags}:** A **View Flags** value that specifies view-related settings for this Web Part. This value can be equal to 0.

**{AllUsers}:** A flag that indicates whether customization or personalization is in effect on this Web Parts page. This value MUST be one of the following.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User not specified; changes apply to all users of this Web Part instance. Also indicates customization mode.</td>
</tr>
<tr>
<td>2</td>
<td>Personalization is in effect; changes apply to the tp_UserId in the Personalization Table.</td>
</tr>
<tr>
<td>3</td>
<td>Personalization is in effect; changes apply to the tp_UserId in the Web Parts Table.</td>
</tr>
</tbody>
</table>

**tp** _Version_: A counter incremented any time a change is made to the schema or other properties of this list and used for internal conflict detection.

**tp** _Cache_: Private data cache for the Web Part. This value can be NULL.

**{Per_tp_Cache}:** Contains the private data cache for documents with a checkout level of 'published'. Otherwise, this MUST be NULL.

**{tp_ListId}:** The List Identifier of the list to which this Web Part refers, enclosed in braces. If not referencing a list, then this value MUST be NULL.

**tp** _Type_: The **Page Type** of this Web Part. This value can be NULL.

**tp** _Source_: Properties of the Web Part, as specified by a compatible HTML editor. This value can be NULL.

**tp** _BaseUrlViewId_: the id of the base view for the list view Web Part. This value can be NULL.

**tp** _View_: Contains implementation-specific XML used when processing this Web Part. If this Web Part is not a view, then this MUST be NULL.

### 3.1.5.19.19 Web Parts Metadata, Nonpersonalized Result Set

The **Web Parts Metadata, Nonpersonalized Result Set** contains the core metadata about the Web Parts appearing on the specified document.

The **Web Parts Metadata, Nonpersonalized Result Set** is part of a series of result sets describing view Web page-related metadata. It MUST return only if the input **@PageView** is 0.

The **Web Parts Metadata, Nonpersonalized Result Set** will not return if **proc_FetchDocForHttpGet** returns a code of 1271, indicating that the site collection was locked.

The **Web Parts Metadata, Nonpersonalized Result Set** MUST contain one row per Web Part.
tp_Id: The Web Part Identifier of the Web Part. This value MUST NOT be NULL.

tp_ZoneId: The name of a Web Part zone. This value can be NULL.

tp_WebPartTypeId: A 16-byte value uniquely identifying the type of the Web Part. MUST NOT be NULL.

tp_Assembly: The assembly name of the implementation of the Web Part.

tp_Class: The fully qualified class name of the implementation of the Web Part.

SolutionId: The solution identifier of the solution. MUST be NULL if there is no solution associated with the Web Part.

Hash: The implementation-specific hash of the contents of the sandboxed solution. This value MUST NOT be NULL.

ValidatorsHash: The implementation-specific hash of the sandboxed solution validators that validated the sandboxed solution. This value MUST NOT be NULL.

Validation>ErrorUrl: The URL to render if the sandboxed solution validator failed validation. This value MUST be NULL if there is no solution associated with the Web Part.

Validation ErrorMessage: The error message string to render if the sandboxed solution validator failed validation. This value MUST be NULL if there is no solution associated with the Web Part.

tp_IsIncluded: If 1, this indicates that the Web Part is visible. MUST NOT be NULL.

tp_FrameState: A value that indicates the frame state of the Web Part. This value MUST be one of the following.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal. The Web Part is displayed in its normal state, with title, content, and placement within the page.</td>
</tr>
<tr>
<td>1</td>
<td>Minimized. The Web Part is collapsed.</td>
</tr>
</tbody>
</table>
**tp_AllUsersProperties**: Properties specified for all users. This value can be NULL.

**tp_PerUsersProperties**: Properties specified for per-user basis. This value can be NULL.

**tp_PartOrder**: Ordinal number that indicates the location of the Web Part in relation to other Web Parts in the same zone. This value can be NULL.

**{tp Flags}**: A View Flags value that specifies view-related settings for this Web Part. This value MUST NOT be NULL.

**{AllUsers}**: A flag that indicates whether customization or personalization is in effect on this Web Parts page. This value MUST be equal to 1, indicating customization.

**tp_Version**: A counter incremented any time a change is made to the schema or other properties of this list and used for internal conflict detection.

**tp_Cache**: Private data cache for the Web Part. This value can be NULL.

**{Per_tp_Cache}**: Private data cache for published documents. This value MUST be NULL.

**{tp_ListId}**: The List Identifier of the list to which this Web Part refers, enclosed in braces. If not referencing a list, this value MUST be NULL.

**tp_Type**: The Page Type of this Web Part. This value can be NULL.

**tp_Source**: Properties of the Web Part as specified by a compatible HTML editor. This value can be NULL.

**tp_BaseViewId**: the id of the base view for the list view Web Part, the value can be NULL.

**tp_View**: An nvarchar(max) value containing implementation-specific XML used when processing this Web Part. If this Web Part is not a view, this MUST be NULL.

### 3.1.5.19.20 List Metadata Result Set

The **List Metadata Result Set** contains the metadata for the lists associated with the Web Parts that are included on the specified document.

The **List Metadata Result Set** returns only if there are such Web Parts (at least one row returns in the previously returned result set: Web Parts Metadata, Personalized, (section 3.1.5.19.18), or Web Parts Metadata, Nonpersonalized (section 3.1.5.19.19)).

The **List Metadata Result Set** is part of a series of result sets describing view Web page-related metadata. It MUST be returned only if the input parameter @PageView is not NULL, and it MUST NOT be returned if proc_FetchDocForHttpGet returns a code of 1271, indicating that the site collection was locked.

The **List Metadata Result Set** MUST return one row for each associated list. The **List Metadata Result Set** is defined in the Common Result Sets List Metadata Result Set (section 2.2.4.14).

### 3.1.5.19.21 List Event Receivers Result Set

The **List Event Receivers Result Set** contains the event receivers registered on the lists associated with the Web Parts that appear on the specified document.

The **List Event Receivers Result Set** MUST return only if there are such lists (the **List Metadata Result Set** (section 2.2.4.14) returns with at least one row).

The **List Event Receivers Result Set** MUST contain one row per event receiver registered. The **List Event Receivers Result Set** can be empty. The **List Event Receivers Result Set** is defined in the Common Result Sets Event Receivers Result Set (section 2.2.4.11).
3.1.5.19.22 List Security Information Result Set

The List Security Information Result Set contains permissions information for the lists associated with the Web Parts that appear on the specified document.

The List Security Information Result Set MUST return only if requested (input parameter @PrefetchListScope is set to 1) and if such lists exist (the List Metadata Result Set (section 2.2.4.14) returns with at least one row).

The List Security Information Result Set returns one row per each unique scope associated with the lists.

<table>
<thead>
<tr>
<th>ListId</th>
<th>uniqueidentifier,</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScopeId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>Acl</td>
<td>varbinary(max),</td>
</tr>
<tr>
<td>AnonymousPermMask</td>
<td>bigint;</td>
</tr>
</tbody>
</table>

**ListId**: The List Identifier of the list.

**ScopeId**: The Scope Identifier of the Permissions Scope that applies to the list. This MUST be set to zero if the list does not have Fine grain permission.

**Acl**: Contains the ACL of the Permissions Scope that applies to the list. This MUST be NULL if **ScopeId** is zero.

**AnonymousPermMask**: Contains the WSS Rights Mask on the list in effect for anonymous users. This MUST be 0 if **ScopeId** is zero.

3.1.5.19.23 SiteCollection Custom Action Result Set

The Site Collection Custom Actions Result Set MUST return 1 row for each custom action defined for the site collection containing the specified document.

The Site Collection Custom Actions Result Set is part of a series of result sets describing view Web page-related metadata. It MUST return only if the input parameter @PageView is not NULL, and it will not return if **proc_FetchDocForHttpGet** returns a return code of 1271, signifying that the site collection was locked.

If there are no custom actions defined for the site collection, this result set MUST NOT return any rows. This result set is defined in the Common Result Sets Custom Actions From Scope Result Set (section 2.2.4.2).

3.1.5.19.24 Site Custom Actions Result Set

The Site Custom Actions Result Set MUST return 1 row for each custom action defined for the site collection containing the specified document.

The Site Custom Actions Result Set is part of a series of result sets describing view Web page-related metadata. It MUST return only if the input parameter @PageView is not NULL, and it will not return if **proc_FetchDocForHttpGet** returns a return code of 1271, signifying that the site collection was locked.

If there are no custom actions defined for the site, this result set MUST NOT return any rows. This result set is defined in the Common Result Sets Custom Actions From Scope Result Set (section 2.2.4.2).

3.1.5.19.25 List Custom Actions Result Set
The **List Custom Action Result Set** contains information about the Custom Actions related to the lists associated with the specified document.

The **List Custom Actions Result Set** is part of a series of result sets describing view Web page-related metadata. It MUST return only if the input parameter @PageView is not NULL, and it will not return if proc_FetchDocForHttpGet returns a return code of 1271, signifying that the site collection was locked.

The **List Custom Actions Result Set** MUST return only if such lists exist (the **List Metadata Result Set** (section 2.2.4.14) returns with at least one row).

If there are no custom actions defined for the list, this result set MUST NOT return any rows. This result set is defined in the Common Result Sets **Custom Actions From Scope Result Set** (section 2.2.4.2).

### 3.1.5.19.26 List Web Parts Result Set

The **List Web Parts Result Set** contains information about the Web Parts related to the lists associated with the specified document.

The **List Web Parts Result Set** MUST return only if such lists exist (the **List Metadata Result Set** (section 2.2.4.14) returns with at least one row).

The **List Web Parts Result Set** MUST contain one row per Web Part registered for each list. The **List Web Parts Result Set** can be empty. The **List Web Parts Result Set** is defined in the Common Result Sets **List Web Parts Result Set** (section 2.2.4.15).

### 3.1.5.19.27 Content Type Order Result Set

The **Content Type Order Result Set** provides the information necessary for the implementation-specific rendering of a list View Web page. The information necessary to render the content types in the configured order, if such exists, is provided in a binary array value.

The **Content Type Order Result Set** MUST return only if @PageView is not NULL, and it will not return if proc_FetchDocForHttpGet returns a code of 1271, indicating that the site collection was locked.

```sql
{CurrentFolderURL}    varchar,
{MetaInfo}            varbinary(max)
```

**{CurrentFolderURL}:** The URL of the current folder as specified in the @CurrentFolderURL input parameter.

**{MetaInfo}:** The information about content type order, embedded in a binary array. If no information is available for the current folder, or if there is no current folder specified, or if the document is not contained in a list, then this parameter MUST be NULL and the column is unnamed. Otherwise, the column MUST be named 'MetaInfo'.

### 3.1.5.19.28 Current Folder Scope Result Set

The **Current Folder Scope Result** set contains scope information about the current folder as specified by the input parameter @CurrentFolderURL. It is used by the front-end Web server to security trim the information rendered to the user in a view Web page based on the user's access.

The Current Folder Scope Result Set MUST return only if @PageView is not NULL, and it MUST NOT return if proc_FetchDocForHttpGet returns a code of 1271, indicating that the site collection was locked. In addition, the **Current Folder Scope Result Set** MUST return only if the document is contained in a list and resides in a folder that is not the list's root folder.
If the **Current Folder Scope Result Set** returns, it MUST return one row.

```sql
{FolderScopeId} uniqueidentifier,
{FolderId} int;
```

**{FolderScopeId}**: The Scope Identifier of the security scope effective for the specified folder.

**{FolderId}**: The integer identifier of the folder document within the containing list.

### 3.1.5.19.29 Navigation Context Security Information Result Set

The **Navigation Context Security Information Result Set** contains Security information about the site (2) containing the specified document and about all sites in its navigation hierarchy.

The **Navigation Context Security Information Result Set** MUST return only upon successful execution, if @PageView is not NULL and the information is not larger than 1,800 bytes.

If the **Navigation Context Security Information Result Set** returns, it MUST return one row for each unique scope in the site's navigation hierarchy, excluding the site's own scope.

The **Navigation Context Security Information Result Set** is defined in the Common Result Sets **Unique Permissions Result Set** (section 2.2.4.26).

### 3.1.5.19.30 NULL Navigation Context Security Information Result Set

The **NULL Navigation Context Security Information Result Set**, with NULL values in three unnamed columns, returns to indicate that the navigation context security information is larger than 1,800 bytes.

The **NULL Navigation Context Security Information Result Set** MUST return only upon successful execution if @PageView is not NULL and the security information about the site (2) or the parent site in the site's navigation hierarchy was larger than 1,800 bytes.

If the **NULL Navigation Context Security Information Result Set** returns, it MUST return one row, as defined in the Common Result Sets **NULL Unique Permissions Result Set** (section 2.2.4.17).

### 3.1.5.19.31 Empty Navigation Context Security Information Result Set

The **Empty Navigation Context Security Information Result Set**, holding zero rows with a single, unnamed NULL column, is returned to indicate that the navigation context security information is not available or is not up-to-date. The **Empty Navigation Context Security Information Result Set** MUST return only upon successful execution, and if neither the **Navigation Context Security Information** (section 3.1.5.19.29) nor the **NULL Navigation Context Security Information** (section 3.1.5.19.30) Result Sets return.

If the **Empty Navigation Context Security Information Result Set** returns, it MUST return zero rows, as defined using T-SQL syntax in the Common Result Sets **Empty Result Set** (section 2.2.4.10).

### 3.1.5.19.32 App Principal Information Result Set

If the @AppPrincipalName parameter is not NULL, the App Principal Info Result Set MUST be returned as defined in section 2.2.4.29.

### 3.1.5.19.33 App Principal Permissions Result Set
The **App Principal Permissions Result Set** is returned to indicate what permissions the specified app principal has for the document as specified in section 2.2.4.30. If the specified app principal has permissions for the specified document, the **App Principal Permissions Result Set** MUST be returned.

### 3.1.5.20 proc_FetchDocForRead

The **proc_FetchDocForRead stored procedure** is invoked to request the metadata information and document stream of a document.

```sql
PROCEDURE proc_FetchDocForRead(
  @DocSiteId uniqueidentifier,
  @DocWebId uniqueidentifier,
  @DocDirName nvarchar(256),
  @DocLeafName nvarchar(128),
  @DocFullUrl nvarchar(260),
  @LooksLikeAttachmentFile bit,
  @GetContent bit,
  @GetWebViewForNormalization bit,
  @bGetContainingList bit,
  @bCheckout bit,
  @GetCurrentMetaInfo bit,
  @GetLinkInfo bit,
  @UserId int,
  @Version int,
  @ChunkSize int,
  @MaxLevel tinyint,
  @StreamPartition tinyint OUTPUT,
  @Level tinyint,
  @UIVersion int OUTPUT,
  @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

**@DocSiteId:** The **Site Collection Identifier** (section 2.2.1.9) of the **site collection** containing the requested document.

**@DocWebId:** The **Site Identifier** (section 2.2.1.11) of the **site (2)** containing the requested document.

**@DocDirName:** The **directory name** of the requested document.

**@DocLeafName:** The **leaf name** of the requested document.

**@DocFullUrl:** This parameter MUST be NULL and MUST be ignored.

**@LooksLikeAttachmentFile:** Specifies whether the requested document appears to the **front-end Web server** to be an **attachment** to a list item. If this flag is set to 1, then this stored procedure MUST determine whether the current user has sufficient permissions to retrieve the requested document as an attachment. In addition, if this flag is set to 1, then the NeedManageListRight column of the **Attachment State Result Set** (section 3.1.5.20.14) will indicate if the ManageLists bit of the **WSS Rights Mask** (section 2.2.2.15) is set.

**@GetContent:** A bit flag specifying whether to return the document stream of the document or not (0).

**@GetWebViewForNormalization:** A bit flag specifying whether to return the subsite of the site (2) specified by **@DocWebId**. If this flag is set to 1, then this procedure MUST return a list of subsites in the **Subsite List Result Set** (section 3.1.5.20.1).
@bGetContainingList: Specifies whether to return the event receivers information about the list (1) containing the requested document. If this flag is set to 1, then this procedure MUST return the Event Receivers Result Set (section 3.1.5.20.6).

@bCheckout: Specifies whether the current user is requesting to check out the document. If this flag is set to 1, and the @LooksLikeAttachmentFile parameter is set to 1, then this stored procedure MUST determine whether the current user has sufficient permissions to check out the requested document as an attachment. The result of this is returned by the Attachment State Result Set (section 3.1.5.20.14).

@GetCurrentMetaInfo: If this parameter is set to 1, then the Document Version Metadata Result Set (section 3.1.5.20.4) MUST return the document's metadata in the {MetaInfo} column. If this parameter is not set to 1, then the result set MUST return NULL as the document's metadata in the {MetaInfo} column.

@GetLinkInfo: Specifies whether to return the link information about the requested document. If this flag is set to 1, then this procedure MUST return the Link Info Single Doc Result Set (section 3.1.5.20.2).

@UserId: The User Identifier (section 2.2.1.1.13) for the current user requesting the information.

@Version: Specifies the user interface (UI) version number of the document that is being requested. A value of -1 specifies the most recent version of the document.

@ChunkSize: Specifies the maximum size, in bytes, of the document content to be returned in the {Content} column in the Document Information and Content (Read) Stream Result Set (section 3.1.5.20.11) and the Document Version Information and Content Stream Result Set (section 3.1.5.20.13). For an uncustomized document, this MUST be NULL. Otherwise, if the data in the {Content} column is larger than the value specified in the @ChunkSize parameter, only the first @ChunkSize bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.

@MaxLevel: A Publishing Level Type (section 2.2.2.6) value that indicates the maximum publishing level of the document to be returned in the @Level parameter if multiple levels of the document are available and the current user is not the owner of the draft or does not have the document checked out.

@StreamPartition: Specifies the stream partition of the document to fetch.

@Level: The Publishing Level Type value of the requested version of the document visible to the current user, returned as an output parameter. This value MUST be returned as NULL if the document does not exist.

@UIVersion: Output parameter. The user interface version number associated with the document. This value MUST be NULL in the case of a document that does not exist, or if the @GetContent flag is not 1.

@RequestGuid: The optional request identifier for the current request.

**Return Values:** This procedure returns an integer return code that MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The document does not exist.</td>
</tr>
</tbody>
</table>

This stored procedure returns a Publishing Level Type value in the parameters as described earlier. This stored procedure MUST return multiple result sets. Some of the result sets MUST NOT be
returned depending upon input parameters or calculations performed in this stored procedure, and all result sets that are returned MUST be sent in the order defined as follows.

3.1.5.20.1 Subsite List Result Set

The Subsite List Result Set contains an unordered list of store-relative URLs for all subsites within the site collection whose parent site (2) is specified in the @DocWebId parameter.

The Subsite List Result Set MUST only be returned if @GetWebListForNormalization is set to 1. The Subsite List Result Set MUST contain one row for each subsite with the specified parent site, and it MUST contain no rows if there are no such subsites.

The Subsite List Result Set is defined in the Common Result Sets URL Result Set (section 2.2.4.27).

3.1.5.20.2 Link Info Single Doc Result Set

The Link Info Single Doc Result Set contains information about links to or within the requested document. Entries are present both for forward links and for backward links. The Link Info Single Doc Result Set MUST only be returned if @GetLinkInfo is set to 1, and MUST contain one row for each link that is referenced.

The Link Info Single Doc Result Set is defined in the Common Result Sets Link Information Result Set (section 2.2.4.13).

3.1.5.20.3 Document Metadata Result Set

The Document Metadata Result Set contains the metadata about the most current version of the requested document visible to the current user.

The Document Metadata Result Set MUST only be returned if @Version is negative. There MUST be one row in the Document Metadata Result Set if the document exists; otherwise, there MUST be no rows.

The Document Metadata Result Set is defined in the Common Result Sets Document Metadata Result Set (section 2.2.4.8).

3.1.5.20.4 Document Version Metadata Result Set

The Document Version Metadata Result Set contains the metadata about the requested version of the document.

The Document Version Metadata Result Set MUST only be returned if @Version is not negative. If the document exists, the Document Version Metadata Result Set MUST contain one row; otherwise, it MUST contain zero rows.

The Document Version Metadata Result Set is defined in section Document Version Metadata Result Set (section 2.2.4.9), located in the Common Result Sets (section 2.2.4).

3.1.5.20.5 NULL Result Set

The NULL Result Set is a placeholder that returns no data.

The NULL Result Set MUST only be returned if @Version is negative, the requested document exists, and @DocWebId is NULL. The NULL Result Set MUST contain zero rows in a schema containing a single unnamed column.

3.1.5.20.6 Event Receivers Result Set
The **Event Receivers Result Set** contains information about the **event receivers** defined for this document.

The **Event Receivers Result Set** MUST only be returned if one of these conditions is true:

- The requested document exists and either @Version is not negative, OR
- @Version is negative and @DocWebId is not NULL

The **Event Receivers Result Set** MUST contain one row for each event receiver registered with an **Event Host Type** (section 2.2.1.2.5) of 3 (list item) for this document.

The **Event Receivers Result Set** is defined in the Common Result Sets **Event Receivers Result Set** (section 2.2.4.11).

### 3.1.5.20.7 List Metadata Result Set

The **List Metadata Result Set** contains the metadata associated with the **list (1)** that contains the requested **document**.

The **List Metadata Result Set** MUST only be returned if the @bGetContainingList parameter is set to 1 and the document is contained within a list. The **List Metadata Result Set** MUST contain one row.

The **List Metadata Result Set** is defined in the Common Result Sets **List Metadata Result Set** (section 2.2.4.14).

### 3.1.5.20.8 Empty List Result Set

The **Empty List Result Set** contains a single, unnamed column with no rows to indicate that the **document** is not contained within a list. The **Empty List Result Set** MUST only be returned if the @bGetContainingList parameter is set to 1 and the document is not contained within a list.

The **Empty List Result Set** is defined in the Common Result Sets **Empty Result Set** (section 2.2.4.10).

### 3.1.5.20.9 Event Receivers Result Set (1)

The **Event Receivers Result Set** contains the **event receivers** associated with the **list (1)** that contains the requested document.

The **Event Receivers Result Set** MUST only be returned if the @bGetContainingList parameter is set to 1 and the document is contained within a list. The **Event Receivers Result Set** MUST contain one row for each event receiver registered with an **Event Host Type** (section 2.2.1.2.5) of 2 (List) for the List.

The **Event Receivers Result Set** is defined in the Common Result Sets **Event Receivers Result Set** (section 2.2.4.11).

### 3.1.5.20.10 Document Information and Content (Read) Metadata Result Set

This result set contains metadata about the **document stream** for the requested **document**.

This result set MUST only be returned if @GetContent is set to 1 and @Version is negative. If the document exists, it MUST contain one row; otherwise, it MUST contain zero rows.

The value returned for the `{HistVersion}` column MUST be 0.

This result set is defined as **Document Content Metadata Result Set** (section 2.2.4.6) in the list of Common Result Sets (section 2.2.4).
3.1.5.20.11 Document Information and Content (Read) Stream Result Set

This result set contains a row for each of the document's stream binary pieces. This result set MUST only be returned if @GetContent is set to "1" and @Version is negative.

This result set MUST return zero or more rows. If the requested document is an uncustomized document, this result set MUST contain 0 rows.

This result set is defined as Document Content Stream Result Set (section 2.2.4.7) in the Common Result Sets (section 2.2.4).

3.1.5.20.12 Document Version Information and Content Result Set

This result set contains metadata about the document stream for the requested document.

This result set MUST only be returned if @GetContent is set to 1 and @Version is not negative. If the document exists, it MUST contain one row; otherwise, it MUST contain zero rows.

The value returned for the {HistVersion} column MUST be the UI version of the requested document version.

This result set is defined as Document Content Metadata Result Set (section 2.2.4.6) in the list of Common Result Sets (section 2.2.4).

3.1.5.20.13 Document Version Information and Content Stream Result Set

This result set contains a row for each of the document's stream binary pieces. This result set MUST only be returned if @GetContent is set to 1 and @Version is not negative.

This result set MUST return zero or more rows.

This result set is defined as Document Content Stream Result Set (section 2.2.4.7) in the Common Result Sets (section 2.2.4).

3.1.5.20.14 Attachment State Result Set

The Attachment State Result Set contains the information about the attachment state of the requested document. The Attachment State Result Set MUST be returned and MUST contain one row.

The Attachment State Result Set is defined using T-SQL syntax, as follows.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>{IsAttachment}</td>
<td>bit</td>
</tr>
<tr>
<td>{NeedManageListRight}</td>
<td>bit</td>
</tr>
<tr>
<td>{Level}</td>
<td>tinyint</td>
</tr>
</tbody>
</table>

{IsAttachment}: This specifies whether the document is associated with an attachment. This value MUST be 1 if the document is an attachment, a list item attachment folder, or the list attachment folder itself. Otherwise, this value MUST be 0. See Attachments Flag (section 2.2.1.2.1) for more information.

{NeedManageListRight}: This specifies whether operations on this document require the user to have the ManageLists bit of the WSS Rights Mask set (section 2.2.2.15). This value MUST be 0 if the document is not stored in a list.

{Level}: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of the document. This value MUST be NULL if the document does not exist.

3.1.5.20.15 Audit Mask Result Set
The Audit Mask Result Set contains the information about the Audit Flags associated with this document. The Audit Mask Result Set MUST be returned and MUST contain a single row of data if the document exists.

The Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).

3.1.5.21 proc_FetchDocForUpdate

The proc_FetchDocForUpdate stored procedure requests document content and metadata information. It also updates the CacheParseId flag for the requested document if the @CacheParse parameter is set to 1.

```
PROCEDURE proc_FetchDocForUpdate(
    @DocSiteId                     uniqueidentifier,
    @DocWebId                      uniqueidentifier,
    @DocDirName                    nvarchar(256),
    @DocLeafName                   nvarchar(128),
    @UserId                        int,
    @Version                       int,
    @GetContent                    bit,
    @GetWebListForNormalization    bit,
    @CacheParse                    bit,
    @GetWebPartInfo                bit,
    @GetFileFormatInfo             bit,
    @bGetContainingList            bit,
    @GetCurrentMetaInfo            bit,
    @LooksLikeAttachmentFile       bit,
    @ChunkSize                     int,
    @StreamPartition               tinyint,
    @Level                         tinyint OUTPUT,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.1.9) of the site collection containing the requested document.

@DocWebId: The Site Identifier (section 2.2.1.1.11) of the site (2) containing the requested document.

@DocDirName: The directory name of the requested document.

@DocLeafName: The leaf name of the requested document.

@UserId: The User Identifier (section 2.2.1.1.13) for the current user requesting the information.

@Version: Specifies the user interface (UI) version number of the document being requested. A value of -1 specifies the most recent version of the document.

@GetContent: A bit flag specifying whether or not to return the document stream of the document. This flag MUST be set to 1 for to return either the Document Information and Content Result Set (section 3.1.5.21.14), the Document Version 1 Information and Content Result Set (section 3.1.5.21.16), or the Document Version 2 Information and Content Result Set (section 2).

@GetWebListForNormalization: A bit flag specifying whether to return the subsite of the site (2) specified by @DocWebId. If this flag is set to 1, then proc_FetchDocForUpdate MUST return a list of child sites in the Subsite List Result Set (section 3.1.5.21.1).

@CacheParse: A bit flag specifying whether to update the CacheParseId flag of the requested document. If this flag is set to 1, then proc_FetchDocForUpdate MUST update the CacheParseId flag in the Docs View (section 2.2.6.3) for the requested document.
@GetWebPartInfo: A bit flag specifying whether to return the Web Parts information for the requested document. If this flag is set to 1, then proc_FetchDocForUpdate MUST return the Web Parts information for the requested document in the Web Part Info Result Set (section 3.1.5.21.9) and the Zone ID Result Set (section 3.1.5.21.9).

@GetFileFormatInfo: If this parameter is set to NULL or 1, then the File Format Metadata Result Set (section 3.1.5.21.10) MUST be returned. Otherwise, the File Format Metadata Result Set MUST NOT be returned.

@bGetContainingList: Specifies whether to return the event receivers information about the list containing the requested document. If this flag is set to 1, then this procedure MUST return the Event Receivers Result Set (section 3.1.5.21.6) for the list.

@GetCurrentMetaInfo: If this parameter is set to 1, then the Document Version Metadata Result Set (section 3.1.5.21.4) MUST return the document's metadata in the {MetaInfo} column. If this parameter is not set to 1, then the Result Set MUST return NULL as the document's metadata in the {MetaInfo} column.

@LooksLikeAttachmentFile: Specifies whether the requested document appears to the front-end Web server to be an attachment to a list item. If this flag is set to 1, then this procedure MUST determine whether the current user has sufficient permissions to retrieve the requested document as an attachment. In addition, if this flag is set to 1, then the NeedManageListRight column of the Attachment State Result Set (section 3.1.5.21.18) will indicate if the ManageLists bit of the WSS Rights Mask (section 2.2.2.15) is set.

@ChunkSize: Specifies the maximum size, in bytes, of the document content to be returned in the {Content} column in the Document Information and Content (Update) Stream Result Set and the Document Version Information and Content Stream Result Set. For an uncustomized document, this MUST be NULL. Otherwise, if the data in the {Content} column is larger than the value specified in the @ChunkSize parameter, only the first @ChunkSize bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.

@StreamPartition: Specifies the stream partition of the document to fetch.

@Level: The Publishing Level Type (section 2.2.2.6) value of the requested version of the document visible to the current user, returned as an output parameter. This value MUST be returned as NULL if the document does not exist.

@RequestGuid: The optional request identifier for the current request.

Return Values: This procedure MUST return an integer return code of 0.

This procedure returns a Publishing Level Type value in the parameters as specified earlier. This procedure MUST return multiple result sets. Some of the result sets MUST NOT be returned, depending upon input parameters or calculations performed in this procedure, and all result sets that are returned MUST be sent in the order defined as follows.

3.1.5.21.1 Subsite List Result Set

The Subsite List Result Set contains an unordered list of store-relative URLs for all subsites within the site collection whose parent site (2) is specified in the @DocWebId parameter.

The Subsite List Result Set MUST only be returned if @GetWebListForNormalization is set to 1. The Subsite List Result Set MUST contain one row for each subsite within the specified parent Site, and MUST contain zero rows if there are no such subsites.

The Subsite List Result Set is defined in the Common Result Sets URL Result Set (section 2.2.4.27).

3.1.5.21.2 ACL and Permission Result Set
The **ACL and Permission Result Set** contains information about the permissions associated with the security scope in effect for the document. The **ACL and Permission Result Set** MUST be returned and MUST contain one row. If the document does not exist, then the values of both columns MUST be NULL. The **ACL and Permission Result Set** is defined in the Common Result Sets **ACL and Permission Result Set** (section 2.2.4.1).

### 3.1.5.21.3 Document Metadata Result Set

The **Document Metadata Result Set** contains the metadata about the most current version of the requested document visible to the current user.

The **Document Metadata Result Set** MUST only be returned if @Version is negative. If the document exists, the **Document Metadata Result Set** MUST contain one row; otherwise, it MUST contain zero rows.

The **Document Metadata Result Set** is defined in the Common Result Sets **Document Metadata Result Set** (section 2.2.4.8).

### 3.1.5.21.4 Document Version Metadata Result Set

The **Document Version Metadata Result Set** contains the metadata about the requested version of the document.

The **Document Version Metadata Result Set** MUST only be returned if @Version is not negative. If the document exists, the **Document Version Metadata Result Set** MUST contain one row; otherwise, it MUST contain zero rows.

The **Document Version Metadata Result Set** is in section **Document Version Metadata Result Set** (section 2.2.4.9), located in the Common Result Sets (section 2.2.4).

### 3.1.5.21.5 NULL Result Set

The **NULL Result Set** is a placeholder that returns no data.

The **NULL Result Set** MUST only be returned if @Version is negative, the requested document exists, and @DocWebId is NULL. The NULL Result Set MUST contain zero rows in a schema containing a single unnamed column.

### 3.1.5.21.6 Event Receivers Result Set

The **Event Receivers Result Set** contains information about the event receivers defined for this document.

The **Event Receivers Result Set** MUST only be returned if the requested document exists in the site specified by @DocWebId. The **Event Receivers Result Set** MUST contain one row for each event receiver registered with an **Event Host Type** (section 2.2.1.2.5) of 3 (list item) for this document.

The result set is defined in the Common Result Sets **Event Receivers Result Set** (section 2.2.4.11).

### 3.1.5.21.7 Link Info Single Doc Fixup Result Set

The **Link Info Single Doc Fixup Result Set** contains information about forward links within the requested document. The **Link Info Single Doc Fixup Result Set** MUST only be returned if Link information was requested by setting @CacheParse to 1.

```sql
LinkDirName       nvarchar(256),
LinkLeafName      nvarchar(128),
LinkId            tinyint,
LinkSecurity      tinyint,
```
LinkDirName: Contains the directory name of the link.
LinkLeafName: Contains the leaf name of the link.
LinkType: The LinkType value of the link in the specified document.
LinkSecurity: A LinkSecurity value specifying whether the scheme of the link is HTTP or HTTPS.
LinkDynamic: A LinkDynamic value that specifies whether this link is one of several special link types.
LinkServerRel: MUST be 1 to indicate that the link URL is a server-relative URL. Otherwise, MUST be 0 to indicate that the link URL is not a server-relative URL.
LinkStatus: A Document Store Type indicating the type of document the link points to. If the link is a forward link for a document that doesn’t exist, then this MUST be NULL. This value MUST be NULL if this link entry refers to a location that exists outside the site collection where the document is stored, or if it refers to a location that could not be verified. See Document Store Type (section 2.2.2.4) for a list of valid values.
PointsToDir: If the link pointed to a directory where a welcome page existed (such as pointing to http://server when a welcome page like http://server/default.aspx existed), then the link is automatically changed to be the URL to the welcome page itself. This bit MUST be set to 1 if this operation has been performed so that it can be distinguished from an explicit link to the welcome page.
WebPartId: If this link corresponds to a Web Part within a Web Part zone, which is therefore logically part of this document, but not present in the HTML source code of this document, then this is the GUID of the Web Part to which the link belongs.
LinkNumber: An ordinal value denoting the relative order of the link within the source of the document, Web Part, or field being processed.
WebId: MUST be NULL.
Search: MUST be NULL.
FieldId: If the link is for a list item field within this document, this is the GUID of the field to which the link belongs.

3.1.5.21.8 Web Part Info Result Set

The Web Part Info Result Set MUST only be returned when @GetWebPartInfo is set to 1. If Web Part data for the requested document exists, one row MUST be returned for each Web Part, otherwise zero rows MUST be returned.
tp_AllUsersProperties          varbinary(max);

**tp_Id**: The Web Part Identifier of the Web Part.

**tp_Level**: Publishing Level Type value of the document.

**tp_Source**: Properties of the Web Part.

**tp_AllUsersProperties**: A list of the XML properties which are common for all users of the Web Part.

### 3.1.5.21.9 Zone ID Result Set

The **Zone ID Result Set** returns a list of **Web Part zone identifiers**. The **Zone ID Result Set** MUST only be returned when `@GetWebPartInfo` is set to 1. If Web Part data for the requested document exists, one row MUST be returned for each Web Part zone; otherwise zero rows MUST be returned.

```plaintext
tp_ZoneID                         nvarchar(64);
```

**tp_ZoneID**: The Web Part zone identifier.

### 3.1.5.21.10 File Format Metadata Result Set

The **File Format Metadata Result Set** returns metadata information about the format of the content of the document. This result set MUST only be returned if the `@GetFileFormatInfo` parameter is set to NULL or 1.

```plaintext
FileFormatMetaInfo              varbinary(max),
FileFormatMetaInfoSize          int
FFMConsistent                   bit
```

@FileFormatMetaInfo: This contains metadata about the format of the content of the document that is to be updated. This can be NULL.

@FileFormatMetaInfoSize: Size in bytes of the metadata returned by the @FileFormatMetaInfo column. This MUST be 0 if the @FileFormatMetaInfo is NULL.

@FFMConsistent: A bit flag specifying whether the file format metadata info is in a consistent state (1) or not (0).

### 3.1.5.21.11 List Metadata Result Set

The **List Metadata Result Set** contains the metadata associated with the list containing the requested document.

The **List Metadata Result Set** MUST only be returned if the `@bGetContainingList` parameter is set to 1 and the document is contained within a list. The **List Metadata Result Set** MUST contain one row.

The **List Metadata Result Set** is defined in the Common Result Sets **List Metadata Result Set** (section 2.2.4.14).

### 3.1.5.21.12 Empty List Result Set

The **Empty List Result Set** contains a single unnamed column with no rows to indicate the document is not contained within a List. The **Empty List Result Set** MUST only be returned if the `@bGetContainingList` parameter is set to 1 and the document is not contained within a List.
The **Empty List Result Set** is defined in the Common Result Sets Empty Result Set (section 2.2.4.10).

### 3.1.5.21.13 Event Receivers Result Set (1)

The **Event Receivers Result Set** contains the event receivers associated with the List which contains the requested document.

The **Event Receivers Result Set** MUST only be returned if the @bGetContainingList parameter is set to 1 and the document is contained within a List. There MUST be one row in the Event Receivers Result Set for each event receiver registered with an Event Host Type (section 2.2.1.2.5) of 2 (List) for the List.

The **Event Receivers Result Set** is defined in the Common Result Sets Event Receivers Result Set (section 2.2.4.11).

### 3.1.5.21.14 Document Information and Content (Update) Metadata Result Set

This result set contains metadata about the document stream for the current version of the requested document.

This result set MUST only be returned if @GetContent is set to 1 and @Version is negative. If the publishing level of the document specified by @Level exists, this result set MUST contain one row; otherwise, this result set MUST contain no rows.

The value returned for the \{HistVersion\} column MUST be 0.

This result set is defined as Document Content Metadata Result Set (section 2.2.4.6) in the Common Result Sets (section 2.2.4).

### 3.1.5.21.15 Document Information and Content (Update) Stream Result Set

This result set contains a row for each of the document's stream binary pieces. This result set MUST only be returned if @GetContent is set to 1 and @Version is negative.

This result set MUST return zero or more rows. If the requested document is an uncustomized document, this result set MUST contain zero rows.

This result set is defined by the Document Content Stream Result Set (section 2.2.4.7) in the Common Result Sets (section 2.2.4).

### 3.1.5.21.16 Document Version Information and Content (Update) Metadata Result Set

This result set contains information about the metadata for the document stream for the requested version of the document.

This result set MUST only be returned if @GetContent is set to 1 and @Version is not negative, and the document exists with the publishing level specified by @Level and with the user interface (UI) version specified by @Version.

This result set MUST contain one row if it is returned. The value returned for the \{HistVersion\} column MUST be the user interface (UI) version of the requested version of the document.

This result set is defined by the Document Content Metadata Result Set (section 2.2.4.6) in the Common Result Sets (section 2.2.4).

### 3.1.5.21.17 Document Version Information and Content (Update) Stream Result Set
This result set contains a row for each of the document's **stream binary pieces**.

This result set MUST only be returned if @GetContent is set to 1 and @Version is not negative, and the document exists with the publishing level specified by @Level and with the user interface (UI) **version** specified by @Version.

This result set MUST contain zero or more rows.

This result set is defined by the Document Content Stream Result Set (section 2.2.4.7) in the Common Result Sets (section 2.2).

### 3.1.5.21.18 Attachment State Result Set

The Attachment State Result Set contains the information about the attachment state of the requested **document**. The Attachment State Result Set MUST be returned and MUST contain one row.

The Attachment State Result Set is defined using T-SQL syntax, as follows.

```sql
{IsAttachment}                bit,
{NeedManageListRight}         bit,
{Level}                       tinyint,
```

**{IsAttachment}**: This specifies whether the document is associated with an attachment. This value MUST be 1 if the document is an attachment, a list item attachment folder, or the list attachment folder itself. Otherwise, this value MUST be 0. See Attachments Flag (section 2.2.1.2.1) for more information.

**{NeedManageListRight}**: This specifies whether operations on this document require the user to have the ManageLists bit of the WSS Rights Mask set (section 2.2.2.15). This value MUST be 0 if the document is not stored in a list.

**{Level}**: A Publishing Level Type (section 2.2.6) value specifying the publishing status of the document. This value MUST be NULL if the document does not exist.

### 3.1.5.22 proc_FetchWelcomeNames

The proc_FetchWelcomeNames stored procedure lists all the names of the default welcome pages used as redirection targets when folders are requested by an HTTP GET in all site collections in the back-end database server.

```sql
PROCEDURE proc_FetchWelcomeNames (                        
    @RequestGuid                   uniqueidentifier = NULL     OUTPUT
);                                                
```

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_FetchWelcomeNames stored procedure MUST return an integer return code of 0.

The proc_FetchWelcomeNames stored procedure MUST return one result set.

### 3.1.5.22.1 Welcome Pages Result Set

The Welcome Pages Result Set returns the list of configured default welcome page names, with one row for each configured welcome page name.
LeafName: A string containing the configured welcome page name in leaf name format; for example, "default.aspx", "default.htm".

3.1.5.23 proc_GenerateNextId

The proc_GenerateNextId stored procedure returns an identifier to be used for a new list item in a specified list, and to increment the value of the identifier returned by the next call to this procedure for the same List.

PROCEDURE proc_GenerateNextId (  
  @SiteId uniqueidentifier,  
  @ListId uniqueidentifier,  
  @NumIds int = 1,  
  @RequestGuid uniqueidentifier = NULL OUTPUT
);

@SiteId: The Site Collection Identifier (section 2.2.1.1.9) of the site collection containing the List.

@ListId: The List Identifier (section 2.2.1.1.5) of the List.

@NumIds: proc_GenerateNextId MUST increment the value of the list item identifier returned by the next call by this value. This value MUST be a positive number (greater than zero). If the parameter is not provided, 1 is used.

@RequestGuid: The optional request identifier for the current request.

Return Values: This procedure returns a positive integer return code, which is the available list item Identifier for the specified List. This procedure MUST NOT return a result set.

3.1.5.24 proc_GetAllRolesForUser

The proc_GetAllRolesForUser stored procedure retrieves the role identifiers for the roles assigned to a set of principals in a given security scope.

PROCEDURE proc_GetAllRolesForUser (  
  @SiteId uniqueidentifier,  
  @ScopeId uniqueidentifier,  
  @PrincipalIdsCsv nvarchar(max),  
  @RequestGuid uniqueidentifier = null OUTPUT,
);

@SiteId: The site collection identifier of the site collection containing the List

@ScopeId: The scope identifier of the security scope for which roles are to be returned.

@PrincipalIdsCsv: The principal (1) identifiers for which roles are to be returned. This parameter MUST be an nvarchar(max) which conforms to the following ABNF:

PrincipalIdsCsv = #PrincipalId  
PrincipalId = 1*DIGIT

Each PrincipalId MUST be greater than 0 and less than 2,147,483,648 when interpreted as a decimal integer.
@RequestGuid: The optional request identifier for the current request.

Result Sets:

This procedure MUST return the User Roles Result Set.

### 3.1.5.24.1 User Roles Result Set

RoleId int NOT NULL,

RoleId: A role identifier assigned to one or more of the principal identifiers specified by the PrincipalId listed by the @PrincipalIdsCsv parameter.

### 3.1.5.25 proc_GetAuditMask

The proc_GetAuditMask stored procedure is invoked to identify Audit Flags (section 2.2.2.1) information for a specified object (a page, file, document, or site collection).

```sql
PROCEDURE proc_GetAuditMask(
    @ItemType                      tinyint,
    @SiteId                        uniqueidentifier,
    @DirName                       nvarchar(256),
    @LeafName                      nvarchar(128),
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

@ItemType: The Audit Item Type of the object.

@SiteId: The Site Collection Identifier for the site collection containing the object. This is used to get the global audit mask.

@DirName: The directory name of the object.

@LeafName: The leaf name of the object.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_GetAuditMask stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>2</td>
<td>An object of the specified Audit Item Type could not be found at the specified location.</td>
</tr>
</tbody>
</table>

The proc_GetAuditMask stored procedure MUST return one result set.

### 3.1.5.25.1 Audit Mask Result Set

The Audit Mask Result Set contains the context-sensitive identifier for the specified object, and the Audit Flags set (section 2.2.2.1) and inherited on that object. The Audit Mask Result Set MUST be returned and MUST contain a single row. The Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).
3.1.5.26 proc_GetAuditMaskOutput

The proc_GetAuditMaskOutput stored procedure is invoked to get Audit Flags (section 2.2.2.1) information about an object (a page, file, document, or site collection).

PROCEDURE proc_GetAuditMaskOutput(
    @ItemType                      tinyint,
    @SiteId                        uniqueidentifier,
    @DirName                       nvarchar(256),
    @LeafName                      nvarchar(128),
    @ld                            uniqueidentifier              OUTPUT,
    @AuditFlags                    int                           OUTPUT,
    @InheritAuditFlags             int                           OUTPUT,
    @GlobalAuditMask               int                           OUTPUT,
    @RequestGuid                   uniqueidentifier = NULL       OUTPUT
);

@ItemType: The Audit Item Type (section 2.2.1.2.2) of the object.

@SiteId: A Site Collection Identifier (section 2.2.1.1.9) for the site collection containing the object. This is used to get the global audit mask information.

@DirName: The directory name of the object.

@LeafName: The leaf name of the object.

@Id: A document identifier identifying the object, returned as an output parameter.

@AuditFlags: An Audit Flags value of the operations to be audited which are set directly on the specified object, returned as an output parameter.

@InheritAuditFlags: An Audit Flags value of the operations to be audited on the specified object which are inherited from its containing list (1), site (2), or site collection, returned as an output parameter.

@GlobalAuditMask: An Audit Flags value of the operations to be audited on the specified object which are inherited from the specified site collection, returned as an output parameter.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_GetAuditMaskOutput stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>2</td>
<td>An object of the specified Audit Item Type could not be found at the specified location.</td>
</tr>
</tbody>
</table>

The proc_GetAuditMaskOutput stored procedure MUST return no result sets.

3.1.5.27 proc_GetContainingList

The proc_GetContainingList stored procedure is invoked to get metadata and event receiver information about the list containing a specified URL.

PROCEDURE proc_GetContainingList(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @Url                           nvarchar(260),
    @Response          uniqueidentifier = NULL       OUTPUT
);
@RequestGuid: The optional request identifier for the current request.

Return Values: This stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>1</td>
<td>The list was found in a different site (2) other than that specified by the @WebId parameter or the site specified by the @WebId parameter was not found.</td>
</tr>
<tr>
<td>3</td>
<td>The list does not exist in the site (2) that was specified by the @WebId parameter.</td>
</tr>
</tbody>
</table>

This stored procedure MUST return one result set if the containing list was not found, and MUST return two result sets if the containing list was found, as defined in the following sections.

3.1.5.27.1 List Metadata Result Set

The List Metadata Result Set contains the metadata associated with the list containing the value in the @Url parameter.

The List Metadata Result Set MUST only be returned if a containing list is found for the value for the @Url parameter and MUST contain one row.

This result set is defined in section List Metadata Result Set (section 2.2.4.14), which can be found in the Common Result Sets (section 2.2.4).

3.1.5.27.2 Empty Result Set

The Empty Result Set MUST only be returned if a containing list is not found for the value in the @Url parameter.

This result set is defined in section Empty Result Set (section 2.2.4.10), which can be found in the Common Result Sets (section 2.2.4).

3.1.5.27.3 Event Receivers Result Set

The Event Receivers Result Set contains information about the event receivers defined for the list that contains the value in the @Url parameter.

The Event Receivers Result Set MUST only be returned if a containing list is found for the value in the @Url parameter. This result set MUST contain one row for each event receiver registered with an Event Host Type (section 2.2.1.2.5) of 2 for the List.

This result set is defined in Event Receivers Result Set (section 2.2.4.11), which can be found in the Common Result Sets (section 2.2.4).
3.1.5.28  proc_GetContainingListCore

The proc_GetContainingListCore stored procedure is invoked to get metadata and event receiver information about the list containing a specified URL.

```sql
PROCEDURE proc_GetContainingListCore ( 
    @SiteId                        uniqueidentifier, 
    @WebId                         uniqueidentifier, 
    @Url                           nvarchar(260), 
    @bReturnListMetadata           bit, 
    @ListId                        uniqueidentifier OUTPUT, 
    @BaseType                      int = NULL OUTPUT, 
    @ExcludedType                  int = NULL OUTPUT, 
    @ListFlags                     bigint = NULL OUTPUT, 
    @bNeedNoThrottle               bit = NULL OUTPUT, 
    @ItemCount                     int = NULL OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.1.9) of the site collection containing the list.

@WebId: The Site Identifier (section 2.2.1.1.11) of the site (2) containing the list.

@Url: The store-relative form URL of a list item or document within the list. The URL is used to derive the location of the containing list. The leaf name part of @Url is ignored, since it could point to a nonexistent document or list item.

@bReturnListMetadata: Specifies whether the List Metadata Result Set (section 3.1.5.28.1) returned by this stored procedure contains any rows. When this parameter is set to 0, the result set MUST be empty.

@ListId: An output parameter that specifies the List Identifier (section 2.2.1.1.5) of the list that contains the specified URL. If the containing list was not found, this parameter MUST be NULL.

@BaseType: An output parameter that specifies the List Base Type (section 2.2.1.2.11) of the list that contains the specified URL. If the containing list was not found, this parameter MUST be NULL.

@ExcludedType: An output parameter that specifies whether the specified URL is within a special folder type in the containing list. The parameter MUST be set to one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The document location is not contained in a special folder.</td>
</tr>
<tr>
<td>1</td>
<td>The document location is, or is contained within, a forms folder: a folder named &quot;Forms&quot; within a document library.</td>
</tr>
<tr>
<td>2</td>
<td>The document location is, or is contained within, a Web image folder: a folder named &quot;_w&quot; within a document library.</td>
</tr>
<tr>
<td>3</td>
<td>The document location is, or is contained within, a thumbnail folder: a folder named &quot;_t&quot; within a document library.</td>
</tr>
<tr>
<td>4</td>
<td>The document location is at the root folder of the list or document library.</td>
</tr>
</tbody>
</table>

@ListFlags: An output parameter that specifies the List Flags (section 2.2.2.5) of the list that contains the specified URL. If the containing list was not found, this parameter MUST be NULL.

@bNeedNoThrottle: An output parameter that specifies whether the list is exempt from resource throttling operations. If the containing list was not found, this parameter MUST be NULL.
@ItemCount: An output parameter that specifies the number of list items that are stored within the list that contains the specified URL. If the containing list was not found, this parameter MUST be NULL.

Return Values: This stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>1</td>
<td>The list was found in a different site (2) other than that specified by the @WebId parameter or the site specified by the @WebId parameter was not found.</td>
</tr>
<tr>
<td>3</td>
<td>The list does not exist in the site (2) that was specified by the @WebId parameter.</td>
</tr>
</tbody>
</table>

This stored procedure MUST return one result set, as defined following.

3.1.5.28.1 List Metadata Result Set

The List Metadata Result Set contains the metadata associated with the list containing the value in the @Url parameter.

The List Metadata Result Set MUST be returned if a containing list is found for the value for the @Url parameter and the @bReturnListMetadata parameter is set to 1. The List Metadata Result Set MUST contain one row.

This result set is defined in section List Metadata Result Set (section 2.2.4.14), which can be found in the Common Result Sets (section 2.2.4).

3.1.5.28.2 Empty Result Set

The Empty Result Set MUST be returned if a containing list is not found for the value in the @Url parameter, or if the @bReturnListMetadata parameter is set to 0.

This result set is defined in section Empty Result Set (section 2.2.4.10), which can be found in the Common Result Sets (section 2.2.4).

3.1.5.29 proc_GetDocsMetaInfo

The proc_GetDocsMetaInfo stored procedure is invoked to request document metadata information for up to ten documents within a specified site (2).
@AttachmentsFlag4         tinyint      = NULL,
@Level4                  tinyint      = NULL,
@DirName5                nvarchar(256) = NULL,
@LeafName5               nvarchar(128) = NULL,
@AttachmentsFlag5        tinyint      = NULL,
@Level5                  tinyint      = NULL,
@DirName6                nvarchar(256) = NULL,
@LeafName6               nvarchar(128) = NULL,
@AttachmentsFlag6        tinyint      = NULL,
@Level6                  tinyint      = NULL,
@DirName7                nvarchar(256) = NULL,
@LeafName7               nvarchar(128) = NULL,
@AttachmentsFlag7        tinyint      = NULL,
@Level7                  tinyint      = NULL,
@DirName8                nvarchar(256) = NULL,
@LeafName8               nvarchar(128) = NULL,
@AttachmentsFlag8        tinyint      = NULL,
@Level8                  tinyint      = NULL,
@DirName9                nvarchar(256) = NULL,
@LeafName9               nvarchar(128) = NULL,
@AttachmentsFlag9        tinyint      = NULL,
@Level9                  tinyint      = NULL,
@DirName10               nvarchar(256) = NULL,
@LeafName10              nvarchar(128) = NULL,
@AttachmentsFlag10       tinyint      = NULL,
@Level10                 tinyint      = NULL,
@RequestGuid             uniqueidentifier = NULL OUTPUT

@DocSiteId: The site collection identifier for the site collection containing the specified documents.

@WebFullUrl: The store-relative URL of the site (2) containing the specified documents.

@GetDocsFlags: A bit mask with a flag that specifies whether to return link information. If this parameter has the bit 0x00000020 set, then link information MUST be returned in the Link Info Result Set (section 3.1.5.29.5).

@UserId: The user identifier for the current user requesting the information.

The next four parameters are duplicated 10 times, with each set of parameters referring to a document to be fetched. Each instance of these individual parameter names is differentiated by a suffix with a value of 1 thru 10, which replaces the placeholder "#" symbol shown below.

@DirName#: The directory name of the specified document. A NULL value signifies that no document is being fetched in this slot, and the next three parameters MUST be ignored.

@LeafName#: The leaf name of the specified document. If @DirName# is not NULL, this MUST NOT be NULL.

@AttachmentsFlag#: An Attachments Flag (section 2.2.1.2.1) value which specifies the type of security checks to be performed by proc_GetDocsMetaInfo based on whether it appears to be an attachment.

@Level#: A Publishing Level Type (section 2.2.2.6) value specifying the maximum publishing level of the current version of the document to be returned to the front-end Web server if multiple current versions of the document are available.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_GetDocsMetaInfo stored procedure returns an integer return code which MUST be 0.
The **proc_GetDocsMetaInfo** stored procedure MUST return multiple result sets:

### 3.1.5.29.1 Individual URL Security Result Set

The **Individual URL Security Result Set** contains security information about the specified document. The **Individual URL Security Result Set** to be used to check if the current user has permission to see this document’s metadata. If the document does not exist, but the specified URL is within a list or document library, security information is returned from the security scope for the specified document location.

One **Individual URL Security Result Set** or **NULL Individual URL Security Result Set** MUST be returned for each document whose corresponding @DirName# parameter is not NULL. If all @DirName# parameters are set to NULL, **Individual URL Security Result Sets** MUST NOT be returned.

Each **Individual URL Security Result Set** MUST only be returned if the specified document location is contained within a List or document library. Otherwise, the **NULL Individual URL Security Result Set** (section 2.2.4.16) MUST be returned instead. If returned, the **Individual URL Security Result Set** MUST contain a single row.

The result set is defined in the Common Result Sets **Individual URL Security Result Set** (section 2.2.4.12).

### 3.1.5.29.2 NULL Individual URL Security Result Set

The **NULL Individual URL Security Result Set** indicates that the specified document location is not contained within a List or document library.

One **Individual URL Security Result Set** or **NULL Individual URL Security Result Set** MUST be returned for each document whose corresponding @DirName# parameter is not NULL. If all @DirName# parameters are set to NULL, **NULL Individual URL Security Result Sets** MUST NOT be returned.

Each **NULL Individual URL Security Result Set** MUST only be returned if the specified document location is not contained within a List or document library. Otherwise, the **Individual URL Security Result Set** (section 2.2.4.12) MUST be returned instead. If returned, the **NULL Individual URL Security Result Set** MUST contain a single row.

The **NULL Individual URL Security Result Set** is defined in the Common Result Sets **NULL Individual URL Security Result Set** (section 2.2.4.16).

### 3.1.5.29.3 Server Time Result Set

The **Server Time Result Set** returns the current time from the back-end database server in UTC. The **Server Time Result Set** MUST be returned. The **Server Time Result Set** MUST only contain a single row. The **Server Time Result Set** is defined in the Common Result Sets **Server Time Result Set** (section 2.2.4.20).

### 3.1.5.29.4 Subsite List Result Set

The **Subsite List Result Set** contains an unordered list of store-relative form URLs for all subsites whose parent site (2) is specified in the @WebFullUrl parameter.

The **Subsite List Result Set** MUST be returned. It MUST contain one row for each subsite with the specified parent Site, and MUST contain no rows if there are no such subsites.

The **Subsite List Result Set** is defined in the Common Result Sets **URL Result Set** (section 2.2.4.27).
3.1.5.29.5 Link Info Result Set

The Link Info Result Set returns information about each forward link from, and each backward link to, the current version of the specified documents within the site collection. The Link Info Result Set MUST only be returned if link information was requested by the @GetDocsFlags parameter. The Link Info Result Set MUST contain one row per link for each specified document.

The Link Info Result Set MUST be ordered by the DocId column, and is defined in the Common Result Sets Single Doc Link Information Result Set (section 2.2.4.21).

3.1.5.29.6 Multiple Document Metadata Result Set

The Multiple Document Metadata Result Set contains the metadata for the specified documents. This result set MUST contain one row for each document where the corresponding @DirName# parameter was not set to NULL, and the rows MUST be ordered by the DocId column.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DocId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>(FullUrl)</td>
<td>nvarchar(260),</td>
</tr>
<tr>
<td>Type</td>
<td>tinyint,</td>
</tr>
<tr>
<td>MetaInfoTimeLastModified</td>
<td>datetime,</td>
</tr>
<tr>
<td>MetaInfo</td>
<td>varbinary(max),</td>
</tr>
<tr>
<td>Size</td>
<td>int,</td>
</tr>
<tr>
<td>TimeCreated</td>
<td>datetime,</td>
</tr>
<tr>
<td>TimeLastModified</td>
<td>datetime,</td>
</tr>
<tr>
<td>ETagVersion</td>
<td>int,</td>
</tr>
<tr>
<td>DocFlags</td>
<td>int,</td>
</tr>
<tr>
<td>tp_Name</td>
<td>nvarchar(38),</td>
</tr>
<tr>
<td>[ListTitle]</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>[CacheParseId]</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>GhostDirName</td>
<td>nvarchar(256),</td>
</tr>
<tr>
<td>GhostLeafName</td>
<td>nvarchar(128),</td>
</tr>
<tr>
<td>tp_Login</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>CheckoutDate</td>
<td>datetime,</td>
</tr>
<tr>
<td>CheckoutExpires</td>
<td>datetime,</td>
</tr>
<tr>
<td>VirusStatus</td>
<td>int,</td>
</tr>
<tr>
<td>VirusInfo</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>SetupPathVersion</td>
<td>tinyint,</td>
</tr>
<tr>
<td>SetupPath</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>SetupPathUser</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>NextToLastTimeModified</td>
<td>datetime,</td>
</tr>
<tr>
<td>UIVersion</td>
<td>int,</td>
</tr>
<tr>
<td>CheckinComment</td>
<td>nvarchar(1023),</td>
</tr>
<tr>
<td>WelcomePageUrl</td>
<td>nvarchar(260),</td>
</tr>
<tr>
<td>WelcomePageParameters</td>
<td>nvarchar(max),</td>
</tr>
<tr>
<td>tp_Flags</td>
<td>bigint,</td>
</tr>
<tr>
<td>Acl</td>
<td>varbinary(max),</td>
</tr>
<tr>
<td>AnonymousPermMask</td>
<td>bigint,</td>
</tr>
<tr>
<td>DraftOwnerId</td>
<td>int,</td>
</tr>
<tr>
<td>Level</td>
<td>tinyint,</td>
</tr>
<tr>
<td>ParentVersion</td>
<td>int,</td>
</tr>
<tr>
<td>TransformerId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>ParentLeafName</td>
<td>nvarchar(128),</td>
</tr>
<tr>
<td>ProgId</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>DocLibRowId</td>
<td>int,</td>
</tr>
<tr>
<td>tp_DefaultWorkflowId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>ListId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>ItemChildCount</td>
<td>int,</td>
</tr>
<tr>
<td>FolderChildCount</td>
<td>int,</td>
</tr>
<tr>
<td>MetaInfoVersion</td>
<td>int,</td>
</tr>
<tr>
<td>{CurVerMetaInfo}</td>
<td>varbinary(max),</td>
</tr>
<tr>
<td>ContentVersion</td>
<td>int,</td>
</tr>
<tr>
<td>{bContentVersionIsDirty}</td>
<td>bit,</td>
</tr>
<tr>
<td>{NextBSN}</td>
<td>bigint,</td>
</tr>
<tr>
<td>{StreamSchema}</td>
<td>byte,</td>
</tr>
</tbody>
</table>
DocId: If the specified document exists, this field MUST be the document identifier; otherwise, this MUST be a generated document identifier value for the specified document.

{FullUrl}: The store-relative form URL for the specified document. This value MUST be NULL if the document does not exist. Otherwise, the size of the nvarchar type returned SHOULD vary depending on whether or not the store-relative form URL has only a directory name, only a leaf name, or both. If it has an empty directory name, it SHOULD be returned as nvarchar(128). If it has an empty leaf name, it SHOULD be returned as nvarchar(256). Otherwise, it MUST be returned as nvarchar(385). Since a front-end Web server cannot determine which type will be returned in advance, it MUST allow for a data type returned as nvarchar(385).

Type: The Document Store Type (section 2.2.4) of this document. This value MUST be NULL if the requested document does not exist.

MetaInfoTimeLastModified: A datetime with a timestamp in UTC format specifying the last time the MetaInfo value of this document was changed. This value MUST be NULL if the document does not exist.

MetaInfo: A metadict for the document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value MUST be NULL if the document does not exist.

Size: The number of bytes in the document stream. This value MUST be NULL if the document does not exist.

TimeCreated: A datetime with a timestamp in UTC format specifying when this document was created. This value MUST be NULL if the document does not exist.

TimeLastModified: A datetime with a timestamp in UTC format specifying when the document was last modified. This corresponds to the TimeCreated (for historical versions) or TimeLastModified (for current versions) of the document. This value MUST be NULL if the document does not exist.

ETagVersion: A counter used for internal conflict detection that is incremented any time a change is made to this document. This value MUST be NULL if the document does not exist.

DocFlags: A Doc Flags value (section 2.2.3) describing the document. This value MUST be NULL if the document does not exist.

{ListType}: A combination of the List Base Type (section 2.2.1.11) and List Server Template (section 2.2.1.12) values of the associated list (1) for this document, where the List Server Template value is multiplied by 256 and added to the value of the List Base Type. This value MUST be NULL if the document does not exist or is not in a list (1).

tp_Name: The identifier of the list (1) that contains this document. This value MUST be NULL if the document does not exist or is not in a list (1).

{ListTitle}: If this document is the root folder of a list (1), this contains the display name of the list (1). This value MUST be NULL if the document does not exist or is not in a list (1).

{CacheParseId}: This value MUST be NULL.

GhostDirName: The directory name as passed to proc_GetDocsMetaInfo (section 3.1.5.29) in the @DirName# parameter.

GhostLeafName: The leaf name as passed to proc_GetDocsMetaInfo in the @LeafName# parameter.
**tp_Login:** If the document exists and is currently checked out, **tp_Login** is the login name of the user to whom it is checked out. In all other cases, this is NULL.

**CheckoutDate:** A datetime with a timestamp in UTC format specifying when this document was checked out. This value MUST be NULL if the document does not exist or is currently checked in.

**CheckoutExpires:** A datetime with a timestamp in UTC format specifying when the short-term lock for this document will expire. If this date is in the past, the document SHOULD be treated as unlocked. This value can be NULL if no short-term lock has been placed on the document.

**VirusStatus:** A **Virus Status** value (section 2.2.1.2.17) specifying the current virus state of this document. This value can be NULL if the document has not been processed by a virus scanner. This value MUST be NULL if the document does not exist.

**VirusInfo:** A string containing a provider-specific message returned by the virus scanner when it last processed the document. This value MUST be NULL if the document does not exist or has not been processed by a virus scanner.

**SetupPathVersion:** For an uncustomized document, this specifies the setup path location that the **SetupPath** fragment is relative to. This value MUST NOT be NULL. The following are valid values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>This is relative to the install location of Windows SharePoint Services 2.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).</td>
</tr>
<tr>
<td>3</td>
<td>This is relative to the install location of Windows SharePoint Services 3.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).</td>
</tr>
<tr>
<td>4</td>
<td>This is relative to the install location of Microsoft SharePoint Foundation 2010 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).</td>
</tr>
<tr>
<td>15</td>
<td>This is relative to the install location of Microsoft SharePoint Foundation 2013 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\15).</td>
</tr>
<tr>
<td>16</td>
<td>The <strong>SetupPath</strong> is relative to the install location of Microsoft SharePoint Server 2016 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\16).</td>
</tr>
</tbody>
</table>

**SetupPath:** For a document that is or has been uncustomized, this contains the setup path fragment relative to the base setup path specified by the **SetupPathVersion** value, where the document stream of this document can be found. This value MUST be NULL if the document does not exist or was never uncustomized.

**SetupPathUser:** If this document is now or once was uncustomized, this contains the login name of the user that created the uncustomized document. This value MUST be NULL if the document does not exist or was never uncustomized.

**NextToLastTimeModified:** The value of **TimeLastModified** from the previous time when the document was last saved. This value MUST be NULL for documents that do not exist.

**UIVersion:** The **UI version** number of the document. This value MUST be NULL for documents that do not exist.

**CheckinComment:** An optional user-supplied description provided when a document is checked in or published. This value MUST be NULL for documents that do not exist or are not checked in.

**WelcomePageUrl:** If this document is a folder, this specifies an optional page to redirect to when the folder is requested with an **HTTP GET** operation. The URL is relative to the URL of the folder itself, and MUST be contained within that folder. Attempts to break out of the folder, such as "/./somepage.aspx", are not valid. This value MUST be NULL if the document does not exist or a welcome page is not specified.
WelcomePageParameters: Contains optional URL parameters to specify as part of the WelcomePageUrl value. These parameters are the URL substring starting at either the query string signifier '?' or the bookmark signifier '#'. This value MUST be NULL if the document that do not exist or parameters are not specified.

tp_Flags: The List Flags value (section 2.2.2.5) for the list (1) that contains this document. This value MUST be NULL for documents that do not exist or are not in a list (1).

Acl: The binary serialization of the WSS ACL Format (section 2.2.3.6) ACL for this document. The WSS ACL is either explicitly defined for the document, or inherited from the parent object of the document. This value MUST be NULL for documents that do not exist.

AnonymousPermMask: A WSS Rights Mask (section 2.2.2.15) indicating the rights granted to an anonymous user, or to a user who has no specific rights to this document. The value MUST be NULL for documents that do not exist.

DraftOwnerId: The User Identifier (section 2.2.1.1.13) of the user that published this document as a draft. This value MUST be NULL if the document does not exist or is not a draft version.

Level: A Publishing Level Type value (section 2.2.2.6) specifying the publishing status of this document. This value MUST be NULL if the document does not exist.

ParentVersion: If the document is a transformed version of another document, this is the UI version value from the parent document. This value MUST be NULL if the document does not exist or is not the product of a document transformation.

TransformerId: If this document is a transformed version of another document, this is the GUID of the agent that performed the transformation. This value MUST be NULL if the document does not exist or is not the product of a document transformation.

ParentLeafName: If the document is a transformed version of another document, this is the leaf name of the original document. The original document MUST be in the same folder as the transformed document. If either is moved, the relationship between the original and transformed document is broken. This value MUST be NULL if the document does not exist or is not the product of a document transformation.

ProgId: Specifies a ProgId or preferred application to open the document. The ProgId value is used to distinguish between different applications that save files with a given file extension (for example, different editors for .HTML or .XML files). This value MUST be NULL if the document does not exist or if a ProgId was not specified when the document was saved.

DoclibRowId: The identifier for a row in a document library for this item. This value MUST be NULL if the document does not exist or is not contained in a list (1).

tp_DefaultWorkflowId: The workflow identifier corresponding to the workflow to be invoked if the document is in a moderated list and the document is submitted for approval as part of a check in. This value MUST be NULL if the document does not exist or is not contained in a list (1) with a configured approval workflow.

ListId: The list identifier of the list (1) that contains the document. This value MUST be NULL if the document does not exist or is not contained in a list (1).

ItemChildCount: The number of non-folder children of the document.

FolderChildCount: The number of folder children of the document.

MetaInfoVersion: A counter used for internal conflict detection that is incremented any time a change is made to the MetaInfo for the document.

{CurVerMetaInfo}: This value MUST be NULL.
**ContentVersion:** A counter used for internal conflict detection that is incremented any time a change is made to the binary contents of this document. This value MUST be NULL if the document does not exist.

{bContentVersionIsDirty}: This value specifies whether the ContentVersion value can be used for conflict detection. If the ContentVersion value can be used for conflict detection or the ContentVersion value is NULL, this value MUST be zero; otherwise, the value MUST be one.

{NextBSN}: The current BLOB sequence number of the document being returned.

{StreamSchema}: The current stream schema of the document being returned.

{InternalVersion}: The internal version number of the document being returned.

### 3.1.5.30 proc_GetLinkInfoSingleDoc

The proc_GetLinkInfoSingleDoc stored procedure provides link information and status for all forward links within a specified document and for all backward links within the specified site collection to the document.

```sql
PROCEDURE proc_GetLinkInfoSingleDoc(
    @DocSiteId uniqueidentifier,
    @DocDirName nvarchar(256),
    @DocLeafName nvarchar(128),
    @Level tinyint,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.1.9) for a site collection containing the document.

@DocDirName: The directory name of the document.

@DocLeafName: The leaf name of the document.

@Level: A Publishing Level Type (section 2.2.2.6) indicating the publishing level of the document.

@RequestGuid: The optional request identifier for the current request.

Return Values: This stored procedure MUST return an integer return code of 0.

This stored procedure MUST return a single result set, as specified in the following section.

#### 3.1.5.30.1 Link Info Single Doc Result Set

The Link Info Single Doc Result Set contains information about links to or within the requested document. Entries are present both for forward links and for backward links. The Link Info Single Doc Result Set MUST be returned, and MUST contain one row for each forward link within the specified document, and one row for each backward link to the document within the specified site collection. The Link Info Single Doc Result Set is defined in section Link Information Result Set (section 2.2.4.13), located in the Common Result Sets (section 2.2.4).

### 3.1.5.31 proc_GetListCheckedOutFiles

The proc_GetListCheckedOutFiles stored procedure is invoked to retrieve a list of all documents with a Document Store Type (section 2.2.2.4) of 0 (File) within a specified list, folder, or its subfolders, which are checked out and do not have checked in draft or published versions.
@SiteId: The site collection identifier of the site collection containing the list.

@ListUrl: The directory name which contains the document(s), preceded with a leading slash ("/").

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_GetListCheckedOutFiles stored procedure returns an integer return code, which MUST be 0.

The proc_GetListCheckedOutFiles stored procedure MUST return one result set.

3.1.5.31.1 Checked Out Files Result Set

The Checked Out Files Result Set returns information in the form defined below about the documents which are checked out. The Checked Out Files Result Set MUST return one row for each document with a Document Store Type (section 2.2.2.4) of 0 (File) in the specified directory which is checked out and does not have a checked in draft or published version.

DirName: The directory name of the directory containing the document(s).

LeafName: The leaf name of the document.

DocLibRowId: The row identifier for the document within the containing document library. If the document is not contained in a list, this value MUST be NULL.

CheckoutUserId: The user identifier for the user who has this document checked out.

{tp_Title}: The display name of the user specified in CheckoutUserId. This parameter MUST NOT be NULL. If display name is unavailable, {tp_Title} MUST contain an empty string.

{tp_Email}: The email address of the user specified in CheckoutUserId. This parameter MUST NOT be NULL. If the email address is unavailable, {tp_Email} MUST contain an empty string.

TimeLastModified: A date/time value in UTC format specifying when the document was last saved.

Size: The size of the document in bytes.

3.1.5.32 proc_GetListFields

The proc_GetListFields stored procedure is invoked to get the mapping of fields in a list. The mapping is represented via an XML string specifying each of the fields in the list.

PROCEDURE proc_GetListFields(
    @SiteId uniqueidentifier,
);
@SiteId: The site collection identifier of the site collection.
@WebId: The site identifier of the site (2) containing the list.
@ListId: The list identifier of the list.
@RequestGuid: The optional request identifier for the current request.

Return Values: This procedure returns an integer return code which MUST be 0. This procedure MUST return a single result set as follows.

### 3.1.5.32.1 Fields Information Result Set

The Fields Information Result Set MUST return a single row holding a single column. The Fields Information Result Set will be returned once, and MUST contain 0 or 1 rows as follows: if no List was found matching the provided @WebId and @ListId parameters, then the result set MUST contain 0 rows. Otherwise, the result set MUST contain one row.

```sql
@SiteId                        uniqueidentifier,
@WebId                         uniqueidentifier,
@ListId                        uniqueidentifier,
@RequestGuid                   uniqueidentifier = NULL     OUTPUT
);
```

#### tp_Fields

Contains a Compressed structure. Uncompressed it contains an implementation-specific version string followed by an XML representation of the field definitions. The field definitions include display and interaction options. The XML MUST conform to the FieldDefinitionDatabaseWithVersion complex type, as specified in section 2.2.7.3.5.

### 3.1.5.33 proc_GetListMetaDataAndEventReceivers

The proc_GetListMetaDataAndEventReceivers stored procedure retrieves information about the metadata, security scopes, Web Parts, and event receivers for a specified list.

```sql
PROCEDURE proc_GetListMetaDataAndEventReceivers(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @ListId                        uniqueidentifier,
    @PrefetchListScope             bit = 0,
    @ThresholdScopeCount           int = 0,
    @PrefetchRelatedFields         bit = 0,
    @PrefetchWebParts              bit = 0,
    @UserId                        int = -1,
    @CurrentFolderUrl              nvarchar(260) = NULL,
    @RequestGuid                   uniqueidentifier = NULL     OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.1.9) for a site collection that contains the site (2) specified by the @WebId parameter.

@WebId: The Site Identifier (section 2.2.1.1.11) for the site (2) containing the list specified by the @ListId parameter.

@ListId: The List Identifier (section 2.2.1.5) for the list.
@PrefetchListScope: Specifies whether to include the Unique Permissions Result Set (section 3.1.5.33.2) or NULL Unique Permissions Result Set (section 3.1.5.33.3) in the returned result sets. If 1, then the Unique Permissions Result Set or NULL Unique Permissions Result Set MUST be returned. Otherwise, the Unique Permissions Result Set and NULL Unique Permissions Result Set MUST NOT be returned.

@ThresholdScopeCount: This parameter MUST be set to a value greater than 0. If the @PrefetchListScope parameter is not set to 0, this parameter MUST specify the maximum number of security scopes returned by the Unique Permissions Result Set.

@PrefetchRelatedFields: If this parameter is not set to 0, the List Related Fields Result Set (section 2.2.4.28) (located in the Common Result Sets (section 2.2.4)) MUST be returned.

@PrefetchWebParts: Specifies whether to include the List Web Parts Result Set (section 3.1.5.33.5) in the returned result sets. If 1, then the List Web Parts Result Set MUST be returned. Otherwise, the List Web Parts Result Set MUST NOT be returned.

@UserId: This value can refer to an existing User Identifier (section 2.2.1.1.13) for the specified site collection, and defaults to -1, specifying that the contents of the List Web Parts Result Set MUST NOT be limited to results for a particular user. When an existing User Identifier is specified, the contents of the List Web Parts Result Set MUST be limited to Web Parts visible to the specified user.

@CurrentFolderUrl: If this value is set to NULL or an empty string this parameter MUST be ignored. Otherwise, it MUST be the root folder URL that the specified list is contained within.

@RequestGuid: The optional request identifier for the current request.

Return Values: This stored procedure returns an integer return code, which MUST be 0. This stored procedure MUST return between two and five result sets, depending upon input parameters. Result sets that are returned will be sent in the order defined in the following sections.

3.1.5.33.1 List Metadata Result Set

The List Metadata Result Set contains the metadata associated with the list. The List Metadata Result Set MUST be returned and MUST contain one row for a valid List Identifier (section 2.2.1.1.5) by the @ListId parameter. If the List Identifier is invalid, the List Metadata Result Set MUST contain zero rows.

The List Metadata Result Set is defined in section List Metadata Result Set (section 2.2.4.14), located in the Common Result Sets (section 2.2.4).

3.1.5.33.2 Unique Permissions Result Set

The Unique Permissions Result Set returns any security scopes associated with list items or folders within the specified list. The Unique Permissions Result Set MUST be returned when the @PrefetchListScope parameter is set to 1, if at least one such security scopes exists. Otherwise, the NULL Unique Permissions Result Set MUST be returned to indicate no such security scopes exist.

The Unique Permissions Result Set MUST contain one row for each security scope defined on any list item or folder within the list, but no more rows than those specified by the @ThresholdScopeCount parameter.

The Unique Permissions Result Set is defined in section Unique Permissions Result Set (section 2.2.4.26), located in the Common Result Sets (section 2.2.4).

3.1.5.33.3 NULL Unique Permissions Result Set

The NULL Unique Permissions Result Set indicates that no unique security scopes exist within the specified list. The NULL Unique Permissions Result Set MUST be returned if the
The @PrefetchListScope parameter is set to 1 and no security scopes are defined for any list items or folders within the specified list.

The NULL Unique Permissions Result Set is defined in section NULL Unique Permissions Result Set (section 2.2.4.17), located in the Common Result Sets (section 2.2.4).

### 3.1.5.33.4 List Event Receivers Result Set

The List Event Receivers Result Set contains information about the event receivers defined for this list. The List Event Receivers Result Set MUST be returned, and MUST contain one row for each event receiver that is registered for this list, or zero rows if no event receivers are registered for this list.

The List Event Receivers Result Set is defined in section Event Receivers Result Set (section 2.2.4.11), located in the Common Result Sets (section 2.2.4).

### 3.1.5.33.5 List Web Parts Result Set

The List Web Parts Result Set contains information about the list Web Parts defined for this site (2). The List Web Parts Result Set MUST be returned if the @PrefetchWebParts parameter is set to 1. The result set MUST contain one row for each Web Part that is registered for this list. If there are no Web Parts registered for this list, then this result set MUST NOT return any rows.

The List Web Parts Result Set is defined in section List Web Parts Result Set (section 2.2.4.15), located in the Common Result Sets (section 2.2.4).

### 3.1.5.33.6 List Related Fields Result Set

The List Related fields Result Set returns information about all of the relationship lookup fields whose target list is the specified list. The List Related fields Result Set MUST return one row for each of the relationship lookup fields whose target list is the specified list. If there are no such fields, then it MUST return zero rows.

The List Related Fields Result Set is defined in section List Related Fields Result Set (section 2.2.4.28), located in the Common Result Sets (section 2.2.4).

### 3.1.5.34 proc_getObject

The proc_getObject stored procedure is invoked to retrieve a single Configuration Object (section 2.2.5.1) from the Configuration Database (section 3.1.1). The proc_getObject is located in the Configuration Database.

```
PROCEDURE proc_getObject(
    @Id uniqueidentifier,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@Id: Contains the GUID for the Id of the Configuration Object to be retrieved.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_getObject stored procedure returns an integer return code, which MUST be 0. The proc_getObject stored procedure MUST return a single result set as follows.

### 3.1.5.34.1 Object Result Set
The **Object Result Set** returns the Configuration Object (section 2.2.5.1) identified by @Id. If @Id is NULL or does not match the value of any configuration objects in the Configuration Database (section 3.1.1), then the **Object Result Set** MUST be returned and MUST return zero rows.

The **Object Result Set** MUST be returned and MUST return one row if the Configuration Object matches the value of @Id.

```sql
Id uniqueidentifier,
ParentId uniqueidentifier,
ClassId uniqueidentifier,
Name nvarchar(128),
Status int,
Version rowversion,
Properties nvarchar(max);
```

**Id**: Contains the GUID for the requested Configuration Object.

**ParentId**: Contains the GUID for the **Parent** (section 2.2.5.4) of the Configuration Object.

**ClassId**: Contains the GUID for the **Class** (section 2.2.5.1) of the Configuration Object.

**Name**: Contains the **Name** (section 2.2.5.3) of the requested Configuration Object.

**Status**: Contains the **Status** (section 2.2.5.5) of the Configuration Object.

**Version**: Contains the **Version** (section 2.2.5.6) of the Configuration Object.

**Properties**: Contains the properties definition of the Configuration Object, which is opaque to the back-end database server.

### 3.1.5.35 proc_getObjectsByBaseClass

The **proc_getObjectsByBaseClass** stored procedure is invoked to return a list of GUIDs for child Configuration Objects (section 2.2.5.1) of the specified parent Configuration Object that inherit from the specified base **Class** (section 2.2.5.1). The **proc_getObjectsByBaseClass** is located in the Configuration Database (section 3.1.1).

```sql
PROCEDURE proc_getObjectsByBaseClass (
    @BaseClassId uniqueidentifier,
    @ParentId uniqueidentifier,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

**@BaseClassId**: Contains the GUID for the base **Class** of the requested Configuration Objects.

**@ParentId**: Contains the GUID for the **Parent** (section 2.2.5.4) Configuration Object of the requested Configuration Objects.

**@RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_getObjectsByBaseClass** stored procedure returns an integer return code which MUST be 0. The **proc_getObjectsByBaseClass** stored procedure MUST return a single result set as follows.

### 3.1.5.35.1 Object ID Result Set

The **Object ID Result Set** contains the GUIDs of the matching Configuration Objects (section 2.2.5.1), if any. The **Object ID Result Set** MUST contain zero rows if no matching Configuration Objects are found, and MUST contain one row for each matching Configuration Object found. The
Object ID Result Set is defined in the Common Result Sets Object ID Result Set (section 2.2.4.18).

3.1.5.36 proc_getObjectsByClass

The proc_getObjectsByClass stored procedure is invoked to retrieve the GUIDs of Configuration Objects (section 2.2.5.1) based upon the Class (section 2.2.5.1.1) type, the parent Configuration Object, and the Name (section 2.2.5.1.3) of the Configuration Object. The proc_getObjectsByClass is located in the Configuration Database (section 3.1.1).

```sql
PROCEDURE proc_getObjectsByClass(  
    @ClassId uniqueidentifier,  
    @ParentId uniqueidentifier,  
    @Name nvarchar(128),  
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@ClassId: Contains the GUID of the Class type of the requested Configuration Objects. MUST NOT be NULL.

@ParentId: Contains the GUID for the Parent (section 2.2.5.1.4) Configuration Object of the requested Configuration Objects.

@Name: Contains the Name of the requested Configuration Object.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_getObjectsByClass stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>50105</td>
<td>Error: Class does not exist.</td>
</tr>
</tbody>
</table>

The proc_getObjectsByClass stored procedure MUST return a single result set as follows.

3.1.5.36.1 Object ID Result Set

The Object ID Result Set contains the GUIDs of the matching Configuration Object (section 2.2.5.1), if any. The Object ID Result Set MUST contain zero rows if no matching Configuration Object are found. A Configuration Object MUST match if @ParentId is NULL or @ParentId is equal to the GUID of the parent of the Configuration Object. In addition, a Configuration Object MUST match if @Name is null OR @Name is equal to the Name of the Configuration Object. The Object ID Result Set is defined in the Common Result Sets Object ID Result Set (section 2.2.4.18).

3.1.5.37 proc_GetSiteFlags

The proc_GetSiteFlags stored procedure is invoked to return the Site Collection Flags (section 2.2.2.9) set for a specified site collection.

```sql
PROCEDURE proc_GetSiteFlags(  
    @WebSiteId uniqueidentifier,  
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```
@WebSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection for which to retrieve Site Collection Flags.

@RequestGuid: The optional request identifier for the current request.

Return Values: This stored procedure MUST return an integer return code of 0.

This stored procedure MUST return the following result set.

### 3.1.5.37.1 Site Collection Flags Result Set

The Site Collection Flags Result Set returns the Site Collection Flags (section 2.2.2.9) for a specified site collection. The Site Flags Result Set will be returned, and MUST contain one row.

```
{SiteCollectionFlags} int;
```

{SiteCollectionFlags}: Contains the Site Collection Flags for the specified site collection. This MUST be NULL if the site collection specified by the @WebSiteId parameter does not exist, is NULL, or is a NULL GUID.

### 3.1.5.38 proc_getSiteMap

The proc_getSiteMap stored procedure is invoked to determine the Site Map (section 3.1.5.38.1) information for a site collection. proc_getSiteMap is located in the Configuration Database (section 3.1.1).

```
PROCEDURE proc_getSiteMap(
   @ApplicationId                 uniqueidentifier,
   @Path                          nvarchar(128),
   @RequestGuid                   uniqueidentifier = NULL     OUTPUT
);
```

@ApplicationId: The Web application identifier containing the site collection specified by the @Path parameter.

@Path: The server-relative path to the root of the site collection. Set @Path to '/' to specify a site collection which is the root of the Web application.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_getSiteMap stored procedure returns an integer return code, which MUST be 0. The proc_getSiteMap stored procedure MUST return a single result set.

### 3.1.5.38.1 Site Map Result Set

The Site Map Result Set returns the Site Map information for the site collection specified by the @ApplicationId and @Path parameter values. The Site Map Result Set MUST return zero rows if the parameters do not match any Site Map information; it MUST return one row if a match is found.

```
Id                             uniqueidentifier,
SubscriptionId                 uniqueidentifier,
DatabaseId                     uniqueidentifier,
RedirectUrl                    nvarchar(512),
Pairing                        tinyint,
SubscriptionName               nvarchar(48),
AppSiteDomainId                varchar(6);
```
Id: Contains the Site Collection Identifier (section 2.2.1.1.9) for the requested site collection.

SubscriptionId: The GUID of the implementation-specific subscription feature for the requested site collection.

DatabaseId: Contains the GUID of the database containing the content of the requested site collection.

RedirectUrl: The URL used to access the specified site collection during an implementation-specific upgrade operation. This value MUST be NULL if the site collection upgrade to the current implementation-specific version is complete.

Pairing: Contains an implementation-specific integer value denoting that a given object is paired with a matching object of a previous version. This parameter is used during an implementation-specific upgrade operation.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>This object is not paired.</td>
</tr>
<tr>
<td>1</td>
<td>This object is paired with an object from a previous major version of the product.</td>
</tr>
</tbody>
</table>

SubscriptionName: The name corresponding to the site subscription identifier of the implementation-specific subscription for the requested site collection.

AppSiteDomainId: The app site domain identifier of the site collection.

3.1.5.39 proc_getSiteMapById

The proc_getSiteMapById stored procedure is invoked to determine complete Site Map (section 3.1.5.38.1) information for a site collection. proc_getSiteMapById is located in the Configuration Database (section 3.1.1).

PROCEDURE proc_getSiteMapById(
    @SiteId                        uniqueidentifier,
    @RequestGuid                   uniqueidentifier = NULL     OUTPUT
);

@SiteId: The Site Collection Identifier (section 2.2.1.1.9) of the site collection to get Site Map (section 3.1.5.38.1) information for.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_getSiteMapById stored procedure returns an integer return code which MUST be 0. The proc_getSiteMapById stored procedure MUST return a single result set.

3.1.5.39.1 Site Map By Id Result Set

The Site Map By Id Result Set returns information about the Web application, database, and URL mapped to the specified site collection. The Site Map By Id Result Set MUST contain one row if the specified site collection identifier exists in the Configuration Database (section 3.1.1) on the back-end database server. Otherwise, it MUST contain zero rows.
Pairing                        tinyint,
HostHeaderIsSiteName           bit,
SubscriptionName               nvarchar(48),
AppSiteDomainId                varchar(6);

**ApplicationId:** The Web application identifier hosting the content of the requested site collection.

**DatabaseId:** The GUID of the database containing the content of the requested site collection.

**Id:** The Site Collection Identifier (section 2.2.1.1.9) for the requested site collection.

**SubscriptionId:** The GUID of the implementation-specific subscription for the requested site collection.

**Path:** The URL used to identify the requested site collection. For site collections that do not use a specifically-configured host domain, the HostHeaderIsSiteName field MUST be 0 and the Path value MUST be the server-relative URL of the root site (2) of the site collection. For site collections using host header mode (that is, site collections identified by host header identifier rather than by server-relative URL path), the HostHeaderIsSiteName field MUST be 1 and the Path value for the site collection MUST be the host name, and MUST include the port number for hosts at non-default ports, and can include the port number for hosts at default ports.

**RedirectUrl:** The URL used to access the specified site collection during an implementation-specific upgrade operation. This value MUST be NULL if the site collection upgrade to the current implementation-specific version number is complete.

**Pairing:** An integer value used during an implementation-specific upgrade operation to denote that a given object is paired with a matching object of a previous implementation-specific version number. The following are valid values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>This object is not paired.</td>
</tr>
<tr>
<td>1</td>
<td>This object is paired with an object from a previous major version of the product.</td>
</tr>
</tbody>
</table>

**HostHeaderIsSiteName:** This value MUST be set to 1 if the requested site collection uses host header mode; otherwise it MUST be 0. The Path field MUST contain the URL for the specified HostHeaderIsSiteName value.

**SubscriptionName:** The name corresponding to the site subscription identifier of the implementation-specific subscription for the requested site collection.

**AppSiteDomainId:** The app site domain identifier of site collection.

### 3.1.5.40 proc_GetTpWebMetaDataAndListMetaData

The proc_GetTpWebMetaDataAndListMetaData stored procedure retrieves metadata for a particular site (2) or list (1).

```sql
PROCEDURE proc_GetTpWebMetaDataAndListMetaData(
    @WebSiteId                     uniqueidentifier,
    @WebId                         uniqueidentifier,
    @Url                           nvarchar(260),
    @ListId                        uniqueidentifier,
    @ViewId                        uniqueidentifier,
    @RunUrlToWebUrl                bit,
    @DGCacheVersion                bigint,
)
```
@WebSiteId: The site collection identifier for a site collection containing the site (2).

@WebId: The site identifier of the site (2) for which metadata is requested, or of the site containing the list for which metadata is requested. This parameter is ignored if the @MetadataFlags has the METADATA_WEB flag set and if values in @WebSiteId and @URL are valid.

@Url: The store-relative form URL of the site (2) or list to retrieve metadata for.

@ListId: The list identifier of the list for which metadata is requested. This parameter is optional and MUST be NULL if no list metadata is required.

@ViewId: The Web Part identifier that is registered for the list specified by the @ListId parameter. If this parameter is set to NULL, then the tp_AllUsersProperties, tp_PerUserProperties, and tp_Cache columns returned by the List Web Parts Result Set (section 2.2.4.15) (found in the Common Result Sets (section 2.2.4)) will be set to NULL.

@RunUrlToWebUrl: A bit flag specifying whether the @URL parameter is for a site (2) or for a list within a site. If this value is set to "1", the value in @URL is for a list, and MUST be converted to a store-relative form URL for the containing site (2).

@DGCacheVersion: The version number of the Domain Group Map Cache in the front-end Web server. This can be compared with the Domain Group Map Cache version number on the back-end database server returned in the Domain Group Cache Versions Result Set (section 2.2.4.4) to determine whether updates are needed. A @DGCacheVersion value of -2 specifies that information about the Domain Group Map Cache is not requested, and the Domain Group Cache BEDS Update Result Set (section 2.2.4.3) and the Domain Group Cache WFE Update Result Set (section 2.2.4.5) MUST NOT be returned.

[SystemId]: The SystemId of the current user requesting the metadata.

@AppPrincipalName: The app principal identifier or the app web domain identifier associated with the current user. If the current user is not associated with an app principal, this parameter MUST be NULL.

@IsHostHeaderAppPrincipalName: If this parameter is set to 1, the @AppPrincipalName parameter MUST be the app principal identifier, else it MUST be the app web domain identifier.

@MetadataFlags: A bit field used to determine the types of metadata returned by this procedure. The only valid values of the @MetadataFlags bits are specified as follows.

<table>
<thead>
<tr>
<th>Symbolic name, Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>METADATA_NONE, 0x00000000</td>
<td>Do not provide any metadata.</td>
</tr>
<tr>
<td>METADATA_FETCH_VIEWS, 0x00000001</td>
<td>Retrieve list views. This flag is set in combination with METADATA_LISTMETADATA_NOFETCH and the @ListId variable and will result in the List Web Parts Result Set (section 3.1.5.40.21).</td>
</tr>
<tr>
<td>METADATA_PREFETCH_SCOPES</td>
<td>Retrieve security scopes. If this flag is set and @ListId is not</td>
</tr>
<tr>
<td>Symbolic name, Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0x00000002</td>
<td>NULL and METADATA_LISTMETADATA_NOFETCH is not set, then the List Unique Permissions Result Set (section 3.1.5.40.18) will be returned.</td>
</tr>
<tr>
<td>METADATA_SAVE_SCOPE_DATA, 0x00000004</td>
<td>Save scope data. This flag is currently unused.</td>
</tr>
<tr>
<td>METADATA FIGURE_TYPE, 0x00000008</td>
<td>Setting this flag in combination with NULL ListID will result in retrieving the ListID values directly from the AllDocs table (section 2.2.6.1).</td>
</tr>
<tr>
<td>METADATA_WEB, 0x00000010</td>
<td>Retrieve information pertaining to the provided site (2) information. This flag is used in combination with other flags to further define the set or sets of information to be retrieved.</td>
</tr>
<tr>
<td>METADATA_URL, 0x00000020</td>
<td>Setting this flag will populate the ListId column for all results, including those which are not list items. When used in combination with a valid @WebSiteId, if @ListId is NULL or doesn't match the list contained within the site (2) identified by @WebSiteId, the Document Metadata Result Set (section 2.2.4.8) is returned.</td>
</tr>
<tr>
<td>METADATA_USERINFOLIST, 0x00000040</td>
<td>Return User information. Setting this flag when METADATA_USERINFOLIST_NOFETCH is not set opens the option for List Metadata Result Set (section 2.2.4.14), List Unique Permissions Result Set (section 3.1.5.40.18), Event Receivers Result Set (section 2.2.4.11), and the List Web Parts Result Set (section 3.1.5.40.21).</td>
</tr>
<tr>
<td>METADATA_USERINFOLIST_FULL, 0x00000080</td>
<td>Return full User information. Setting this flag in combination with METADATA_USERINFOLIST_NOFETCH when METADATA_USERINFOLIST_NOFETCH is not set, along with a valid Site Identifier in @WebSiteId, where @ListId is NULL or does not match a list identifier contained within the site (2) specified by @WebSiteId, returns the Event Receivers Result Set (section 3.1.5.40.20) and the List Web Parts Result Set (section 3.1.5.40.21).</td>
</tr>
<tr>
<td>METADATA_WEB_PROP, 0x00000100</td>
<td>Return site data. Setting this flag in combination with METADATA_WEB will retrieve the Site Metainfo Result Set (section 2.2.4.25).</td>
</tr>
<tr>
<td>METADATA_WEB_FEATURES, 0x00000200</td>
<td>Return site features. Setting this flag in combination with METADATA_WEB will retrieve the Site Feature List Result Set (section 2.2.4.23).</td>
</tr>
<tr>
<td>METADATA_WEB_NAVSTRUCT, 0x00000400</td>
<td>Return navigation structure. Setting this flag in combination with METADATA_WEB will retrieve the Site Unique Permissions Result Set (section 3.1.5.40.10).</td>
</tr>
<tr>
<td>METADATA_LISTMETADATA_NOFETCH, 0x00000800</td>
<td>Do not return list metadata. Setting this flag will suppress the retrieval of the List Metadata Result Set (section 3.1.5.40.16), Unique Permissions Result Set (section 3.1.5.40.23), and the Event Receivers Result Set (section 3.1.5.40.20) using the List identification parameters.</td>
</tr>
<tr>
<td>METADATA_USERINFOLIST_NOFETCH, 0x00001000</td>
<td>Do not return User information. Setting this flag will suppress the retrieval of the List Metadata Result Set (section 3.1.5.40.16), Unique Permissions Result Set (section 2.2.4.26), Event Receivers Result Set (section 3.1.5.40.20), and the List Web Parts Result Set (section 3.1.5.40.21) using the User identification parameters.</td>
</tr>
</tbody>
</table>
Symbolic name, Value | Description
--- | ---
METADATA_URL_REALY, 0x00002000 | Return single document metadata. Setting this flag will return the Document Metadata Result Set (section 3.1.5.40.27).
METADATA_WEB_WELCOMEPAGE, 0x00004000 | Redirect welcome page. Setting this flag in combination with METADATA_WEB will retrieve the Redirect URL Result Set (section 3.1.5.40.13).
METADATA_PREFETCHRELATEDFIELDS, 0x00008000 | Return the List Related Fields Result Set (section 2.2.4.28) if the @MetadataFlags parameter does not contain METADATA_LISTMETADATA_NOFETCH.

@ThresholdScopeCount: This parameter MUST be set to a value greater than 0. If the @MetadataFlags parameter does not contain METADATA_LISTMETADATA_NOFETCH and contains METADATA_PREFETCH_SCOPES, this parameter MUST specify the maximum number of security scopes returned by the List Unique Permissions Result Set (section 3.1.5.40.18).

@CurrentFolderUrl: If this value is set to NULL or an empty string, this parameter MUST be ignored. Otherwise, it MUST be the root folder URL that the specified list is contained within.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_GetTpWebMetaDataAndListMetaData stored procedure returns an integer return code, which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful operation.</td>
</tr>
<tr>
<td>1</td>
<td>Cross Site attack detected.</td>
</tr>
<tr>
<td>2</td>
<td>The attempt to retrieve a redirected URL failed.</td>
</tr>
<tr>
<td>3</td>
<td>The specified URL was not found.</td>
</tr>
<tr>
<td>1168</td>
<td>The site collection specified by @WebSiteId was not found.</td>
</tr>
<tr>
<td>1271</td>
<td>Access to the site (2) is blocked because the site collection is locked.</td>
</tr>
</tbody>
</table>

Result Sets: The proc_GetTpWebMetaDataAndListMetaData stored procedure returns up to 30 result sets, depending on the state of the input parameters.

3.1.5.40.1 Web Url Result Set

The Web Url Result Set contains the store-relative form URL of the root of the Site which contains the @Url parameter.

The Web Url Result Set MUST be returned when @MetadataFlags has the METADATA_WEB flag set, @RunUrlToWebUrl is set to 1, and @WebSiteId is a valid Site Collection Identifier, and MUST contain one row of data. If @MetadataFlags has the METADATA_WEB flag set, @RunUrlToWebUrl is set to 1, and @WebSiteId is not found, proc_GetTpWebMetaDataAndListMetaData MUST NOT return any result sets.

```
{WebUrl}       nvarchar(260);
```

{WebUrl}: The store-relative form URL of the Site containing the @Url parameter.

3.1.5.40.2 Domain Group Cache Versions Result Set
The **Domain Group Cache Versions Result Set** contains information about the version numbers associated with the Domain Group Map Caches on the front-end Web server and on the back-end database server for the specified site collection.

The **Domain Group Cache Versions Result Set** MUST be returned when `@Url` specifies a valid location within or of a site (2) and `@MetadataFlags` has the `METADATA_WEB` flag set, and MUST contain one row of version number data. If `@MetadataFlags` has the `METADATA_WEB` flag set, and `@Url` does not specify either a valid site (2), or if `@RunUrlToWebUrl` is set to 1, a valid location within a site (2), then `proc_GetTpWebMetaDataAndListMetaData` MUST NOT return this or any further result sets.

If the specified `@DGCacheVersion` value is -2, then all columns returned MUST have the value -2, indicating that the value MUST NOT be used for comparison. The **Domain Group Cache Versions Result Set** is defined in the Common Result Sets Domain Group Cache Versions Result Set (section 2.2.4.4).

### 3.1.5.40.3 Domain Group Cache BEDS Update Result Set

The **Domain Group Cache BEDS Update Result Set** contains information to be used in recomputing the domain group map cache, which contains the mapping of domain groups to the **security groups** they are members of. The presence of the **Domain Group Cache BEDS Update Result Set** means the database's copy of the domain group map cache is out of date and MUST be recomputed to ensure that proper security checks can be made.

The **Domain Group Cache BEDS Update Result Set** MUST be returned only if the **Domain Group Cache Versions Result Set** is returned, `@DGCacheVersion` does not contain "-2", and the value of `RealVersion` is greater than the `CachedVersion` in the results in the **Domain Group Cache Versions Result Set**.

The **Domain Group Cache BEDS Update Result Set** schema is defined in the Common Result Sets Domain Group Cache BEDS Update Result Set (section 2.2.4.3).

### 3.1.5.40.4 Domain Group Cache WFE Update Result Set

The **Domain Group Cache WFE Update Result Set** contains the binary data needed to refresh the domain group map cache. The presence of the **Domain Group Cache WFE Update Result Set** means the domain group map cache on the database is up to date, and the front-end Web server cache can be refreshed if necessary.

The **Domain Group Cache WFE Update Result Set** MUST be returned only if the **Domain Group Cache Versions Result Set** is returned, and `@DGCacheVersion` does not contain "-2", and the value of `RealVersion` is less than or equal to `CachedVersion` in the results in **Domain Group Cache Versions Result Set**. The **Domain Group Cache WFE Update Result Set** MUST contain one row of data.

The **Domain Group Cache WFE Update Result Set** schema is defined in the Common Result Sets Domain Group Cache WFE Update Result Set (section 2.2.4.5).

### 3.1.5.40.5 Site Metadata Result Set

The **Site Metadata Result Set** contains specialized site metadata. The **Site Metadata Result Set** MUST be returned when the **Domain Group Cache Versions Result Set** is returned, and it MUST contain one row.

The **Site Metadata Result Set** is defined in the Common Result Sets Site Metadata Result Set (section 2.2.4.24).

### 3.1.5.40.6 Site Collection Event Receivers Result Set
The **Site Collection Event Receivers Result Set** contains information about the event receivers defined for the site collection specified by the `@WebSiteId` parameter. The **Site Collection Event Receivers Result Set** MUST be returned when execution is successful, and the **List Metadata Result Set** has been returned.

The **Site Collection Event Receivers Result Set** MUST contain one row for each event receiver with an **Event Host Type** (section 2.2.1.2.5) of 0 registered for the site collection specified by the `@WebSiteId` parameter.

The **Site Collection Event Receivers Result Set** is defined in **Event Receivers Result Set** (section 2.2.4.11), located in the Common Result Sets (section 2.2.4).

### 3.1.5.40.7 Site Event Receivers Result Set

The **Event Receivers Result Set** contains information about the event receivers defined for this site (2). The **Event Receivers Result Set** MUST be returned when the **Domain Group Cache Versions Result Set** is returned, and it MUST contain one row for each event receiver registered for this site (2). If this **Event Receivers Result Set** is returned, and the site collection specified by `@WebSiteId` is locked, then **proc_GetTpWebMetaDataAndListMetaData** MUST NOT return any further result sets.

The **Event Receivers Result Set** is defined in the Common Result Sets **Event Receivers Result Set** (section 2.2.4.11).

### 3.1.5.40.8 Site MetaInfo Result Set

The **Site MetaInfo Result Set** contains the metadata of the site (2). The **Site MetaInfo Result Set** MUST be returned when execution is successful, and `@MetadataFlag` has the **METADATA_WEB** flag set and the **METADATA_WEB_PROP** flag set, and MUST contain one row if the `@WebId` parameter is valid. If the `@MetadataFlag` has the **METADATA_WEB** flag set and the **METADATA_WEB_PROP** flag set, and the site (2) specified by `@WebId` does not exist, then the result set MUST contain no rows, and **proc_GetTpWebMetaDataAndListMetaData** MUST NOT return any further result sets.

The **Site MetaInfo Result Set** is defined in the Common Result Sets **Site MetaInfo Result Set** (section 2.2.4.25).

### 3.1.5.40.9 Site Feature List Result Set

The **Site Feature List Result Set** contains the list of **Feature Identifiers** (section 2.2.1.1.4) of the site (2). The **Site Feature List Result Set** MUST be returned and MUST contain one row for each feature when execution is successful, and all of the following conditions are met:

- `@MetadataFlag` has the **METADATA_WEB** flag set
- `@MetadataFlag` has the **METADATA_WEB_FEATURES** flag set
- `@WebSiteId` is valid
- `@WebId` is valid

The **Site Feature List Result Set** MUST be returned twice with the same schema, once for the default features for the site collection, and again with the features for the specified site (2).

The **Site Feature List Result Set** is defined in the Common Result Sets **Site Feature List Result Set** (section 2.2.4.23).

### 3.1.5.40.10 Site Unique Permissions Result Set
The **Unique Permissions Result Set** contains security scope and WSS ACL (section 2.2.3.6) information of the site (2) specified by @WebId. The **Unique Permissions Result Set** MUST be returned when execution is successful, and all of the following conditions are met:

- @MetadataFlags has the METADATA_WEB flag set
- @MetadataFlags has the METADATA_WEB_NAVSTRUCT flag set
- The specified site (2) contains cached navigation scope information.

The **Unique Permissions Result Set** is defined in the Common Result Sets **Unique Permissions Result Set** (section 2.2.4.26).

### 3.1.5.40.11 Site NULL Unique Permissions Result Set

The **NULL Unique Permissions Result Set** MUST be returned when execution is successful, and all of the following conditions are met:

- @MetadataFlags has the METADATA_WEB flag set
- @MetadataFlags has the METADATA_WEB_NAVSTRUCT flag set
- The site (2) specified by @WebId or its parent site's cached navigation scope is over 900 characters long.

The **NULL Unique Permissions Result Set** is defined in the Common Result Sets **NULL Unique Permissions Result Set** (section 2.2.4.17).

### 3.1.5.40.12 Empty Result Set

An **Empty Result Set** containing no rows MUST be returned when execution is successful, and all of the following conditions are met:

- @MetadataFlags has the METADATA_WEB flag set
- @MetadataFlags has the METADATA_WEB_NAVSTRUCT flag set
- The site (2) specified by @WebId does not contain cached navigation scope information, or the site is marked as dirty.

The **Empty Result Set** is defined in the Common Result Sets **Empty Result Set** (section 2.2.4.10).

### 3.1.5.40.13 Redirect Url Result Set

The **Redirect Url Result Set** contains redirect URL information generated from the @Url parameter. If execution is successful, and @MetadataFlags has the METADATA_WEB flag set and the METADATA_WEB_WELCOMEPAGE flag set, and the site location specified by @Url has a Redirect Type (section 2.2.1.2.15) of 255 (None), then proc_GetTpWebMetaDataAndListMetaData MUST NOT return this or any further result sets. Otherwise, the **Redirect Url Result Set** MUST be returned when execution is successful, and all of the following conditions are met:

- @MetadataFlags has the METADATA_WEB flag set
- @MetadataFlags has the METADATA_WEB_WELCOMEPAGE flag set,
- The site location has been marked as a welcome page.

The **Redirect Url Result Set** MUST contain one row with the Redirect URL and related information for the specified site location.
### 3.1.5.40.14 No Welcome Redirect Url Result Set

The **No Welcome Redirect Url Result Set** contains redirect URL information generated from the @Url parameter. The **No Welcome Redirect Url Result Set** MUST be returned when execution is successful, and all of the following conditions are met:

- @MetadataFlags has the METADATA_WEB flag set
- @MetadataFlags has the METADATA_WEB_WELCOMEPAGE flag set
- The site location specified by @Url has not been marked as a welcome page, that is, it has a RedirectType that is not 0

The **No Welcome Redirect Url Result Set** will return exactly one row with the Redirect URL for this folder.

```sql
{RedirectType} tinyint,
{RedirectUrl} nvarchar(260),
{WelcomePageParameters} nvarchar(max);
```

**{RedirectType}**: Type of item this URL is redirected to. See [Redirect Type](#) for all possible values for this column.

**{RedirectUrl}**: The *store-relative URL* generated from the provided @Url parameter.

**WelcomePageParameters**: The URL parameters for the welcome page found with this redirect URL. This column can contain a query string starting with "?" or a hash parameter starting with "#" (or both).

### 3.1.5.40.15 List Identifier Result Set

The **List Identifier Result Set** contains identifying information for the file, list or list item specified by @Url. If execution is successful, and @ListId is NULL, @MetadataFlags has the METADATA FIGURE TYPE flag set and the site (2) containing @Url is not the site specified by @WebId, then **proc_GetTpWebMetaDataAndListMetaData** MUST NOT return this or any further result sets. Otherwise, the **List Identifier Result Set** MUST be returned when execution is successful, and all of the following conditions are met:

- @ListId is NULL and
- @MetadataFlags has the METADATA FIGURE TYPE flag set and
- @WebId is a valid site (2) and contains the file, list or list item specified by @Url

```sql
{ListIdSelected} uniqueidentifier,
{TypeSelected} int,
```

**{RedirectType}**: Type of item this URL is redirected to. See the Redirect Type ([section 2.2.1.2.15](#)) for all possible values for this column.

**{RedirectUrl}**: The complete redirect URL generated from the provided @Url parameter.

**{WelcomePageParameters}**: This MUST be NULL and can be ignored.
{ListIdSelected}: The List Identifier of the specified list, or of the list containing the specified list item. This value MUST be NULL if @Url does not specify a list or list item.

{TypeSelected}: Item type value for the specified list. The value of TypeSelected MUST be one of the RedirectType values specified in Redirect Type (section 2.2.1.2.15).

{ItemIdSelected}: Document Library Row Identifier of the list item taken from the Docs View (section 2.2.6.3) for the list. This value MUST be NULL if @Url does not specify a list item.

3.1.5.40.16 List Metadata Result Set

The List Metadata Result Set contains the metadata associated with the list specified by @ListId or if @ListId is NULL, the containing List specified by @Url. The List Metadata Result Set MUST be returned when execution is successful, and @ListId specifies a list or @Url specifies a list or list item, and @MetadataFlags does not have the METADATA_LISTMETADATA_NOFETCH flag set. The List Metadata Result Set MUST contain one row if @WebId specifies a valid site (2) containing the specified list.

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.4.14).

3.1.5.40.17 List Related Fields Result Set

The List Related fields Result Set returns information about all the relationship lookup fields whose target list is the specified list. The List Related fields Result Set MUST return one row for each of the relationship lookup fields whose target list is the specified list. If there are no such fields, then it MUST NOT return any rows.

The List Related Fields Result Set is defined in section List Related Fields Result Set (section 2.2.4.28), located in the Common Result Sets (section 2.2.4).

3.1.5.40.18 List Unique Permissions Result Set

The Unique Permissions Result Set contains all security scope and WSS ACL (section 2.2.3.6) information from the specified list. The Unique Permissions Result Set MUST be returned when execution is successful, the List Metadata Result Set has been returned, and all of the following conditions are met:

- @MetadataFlags has the METADATA_PREFETCH_SCOPES flag set and
- @MetadataFlags has the METADATA_LISTMETADATA_NOFETCH flag not set

The Unique Permissions Result Set is defined in Unique Permissions Result Set (section 2.2.4.26).

3.1.5.40.19 List NULL Unique Permissions Result Set

The NULL Unique Permissions Result Set MUST be returned when execution is successful, the List Metadata Result Set has been returned, and all of the following conditions are met:

- @MetadataFlags has the METADATA_PREFETCH_SCOPES flag set
- @MetadataFlags has the METADATA_LISTMETADATA_NOFETCH flag not set
- Permissions associated with the specified list do not exist
The **NULL Unique Permissions Result Set** is defined in Unique Permissions Result Set (section 2.2.4.26).

### 3.1.5.40.20 Event Receivers Result Set

The **Site Event Receivers Result Set** contains information about the event receivers defined for the site (2) containing the list specified by the @ListId parameter, or if the @ListId parameter is NULL, the containing list specified by the @Url parameter. The Site Event Receivers Result Set MUST be returned when execution is successful, and the List Metadata Result Set has been returned.

The Site Event Receivers Result Set MUST contain one row for each event receiver with an Event Host Type (section 2.2.1.2.5) of 2 registered for the site (2) specified by @WebId.

The Site Event Receivers Result Set is defined in aa (section 2.2.4.11), located in the Common Result Sets (section 2.2.4).

### 3.1.5.40.21 List Web Parts Result Set

The **List Web Parts Result Set** contains information about the List Web Parts defined for the list specified by @ListId, or if @ListId is NULL, the containing list specified by @Url. The List Web Parts Result Set MUST be returned when execution is successful, the List Metadata Result Set has been returned, and @MetadataFlags has the METADATA_FETCH_VIEWS flag set.

The List Web Parts Result Set MUST contain one row for each Web Part registered for this List.

The List Web Parts Result Set is defined in the Common Result Sets List Web Parts Result Set (section 2.2.4.15).

### 3.1.5.40.22 List Metadata Result Set (1)

The **List Metadata Result Set** contains the metadata associated with a specified user information list. The List Metadata Result Set MUST be returned when execution is successful, and all of the following conditions are met:

- @WebSiteId contains a valid Site Collection Identifier and the site collection has a user information list

- The List Identifier specified by @ListId, or if @ListId is NULL, the List Identifier of the list at (or that contains) the location specified by @Url doesn't match the List Identifier for the user information list

- @MetadataFlags has the METADATA_USERINFOFLIST flag set, or @Url specifies a list or a location within a list, and

- @MetadataFlags doesn't have the METADATA_USERINFOFLIST_NOFETCH flag set.

The List Metadata Result Set MUST contain one row if it is returned.

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.4.14).

### 3.1.5.40.23 Unique Permissions Result Set

The **Unique Permissions Result Set** contains all security scope and WSS ACL (section 2.2.3.6) information from a specified user information list. The Unique Permissions Result Set MUST be returned when execution is successful, and all of the following conditions are met:

- @WebSiteId contains a valid Site Collection Identifier and the site collection has a user information list
The **List Identifier** specified by @ListId, or if @ListId is NULL, the **List Identifier** of the list at (or that contains) the location specified by @Url doesn't match the **List Identifier** for the user information list

- @MetadataFlags has the `METADATA_USERINFOLIST` flag set, or @Url specifies a list or a location within a list, and
- @MetadataFlags doesn't have the `METADATA_USERINFOLIST_NOFETCH` flag set
- @MetadataFlags has the `METADATA_PREFETCH_SCOPES` flag set

The **Unique Permissions Result Set** is defined in **Unique Permissions Result Set** (section 2.2.4.26).

### 3.1.5.40.24 NULL Unique Permissions Result Set

The **NULL Unique Permissions Result Set** MUST be returned when execution is successful, and all of the following conditions are met:

- @WebSiteId contains a valid **Site Collection Identifier** and the site collection has a user information list
- The **List Identifier** specified by @ListId, or if @ListId is NULL, the **List Identifier** of the list at (or that contains) the location specified by @Url doesn't match the **List Identifier** for the user information list
- @MetadataFlags has the `METADATA_USERINFOLIST` flag set, or @Url specifies a list or a location within a list, and
- @MetadataFlags doesn't have the `METADATA_USERINFOLIST_NOFETCH` flag set
- @MetadataFlags has the `METADATA_PREFETCH_SCOPES` flag set
- No unique permissions exist for the root **site (2)** of the site collection.

The **NULL Unique Permissions Result Set** is defined in the in the Common Result Sets **NULL Unique Permissions Result Set** (section 2.2.4.17).

### 3.1.5.40.25 Event Receivers Result Set (1)

The **Event Receivers Result Set** contains information about the event receivers defined for the **site (2)** containing the specified user information list. The **Event Receivers Result Set** MUST be returned when execution is successful, and all of the following conditions are met:

- @WebSiteId contains a valid **Site Collection Identifier** and the site collection has a user information list
- The **List Identifier** specified by @ListId, or if @ListId is NULL, the **List Identifier** of the list at (or that contains) the location specified by @Url doesn't match the **List Identifier** for the user information list
- @MetadataFlags has the `METADATA_USERINFOLIST` flag set, or @Url specifies a list or a location within a list
- @MetadataFlags has the `METADATA_USERINFOLIST_NOFETCH` flag set
- @MetadataFlags has the `METADATA_PREFETCH_SCOPES` flag set

The **Event Receivers Result Set** MUST contain one row for each event receiver with an **Event Host Type** (section 2.2.1.2.5) of 2 (List) that is registered for the root site (2) of the site collection specified by @WebSiteId.
The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.4.11).

3.1.5.40.26 List Web Parts Result Set (1)

The List Web Parts Result Set contains information about the Web Parts defined for the specified user information list. The List Web Parts Result Set MUST be returned if execution is successful, and all the following conditions are met:

- @WebSiteId contains a valid Site Collection Identifier and the site collection has a user information list
- The List Identifier specified by @ListId, or if @ListId is NULL, the List Identifier of the list at (or that contains) the location specified by @Url doesn't match the List Identifier for the user information list
- @MetadataFlags has the METADATA_USERINFOLIST flag set, or @Url specifies a list or a location within a list
- @MetadataFlags has the METADATA_USERINFOLIST_FULL flag set
- @MetadataFlags doesn't have the METADATA_USERINFOLIST_NOFETCH flag set

The List Web Parts Result Set MUST contain one row for each Web Part registered for this List.

The List Web Parts Result Set is defined in the Common Result Sets List Web Parts Result Set (section 2.2.4.15).

3.1.5.40.27 Document Metadata Result Set

The Document Metadata Result Set contains the metadata for a single document when the provided @Url parameter refers to a single document. The Document Metadata Result Set MUST be returned when execution is successful, and either of the following conditions sets are met:

- @MetadataFlags has the METADATA_URL flag set
- The document is not contained within a list

OR

- @MetadataFlags has the METADATA_URL_REALLY flag set

The Document Metadata Result Set MUST return one row when @Url is valid. The Document Metadata Result Set MUST be ordered by the Id column, and is defined in the Common Result Sets Document Metadata Result Set (section 2.2.4.8).

3.1.5.40.28 NULL Result Set

The NULL Result Set returns no data. The NULL Result Set MUST be returned if the Document Metadata Result Set is not empty, otherwise the NULL Result Set MUST NOT be returned. The NULL Result Set MUST return zero rows in a schema containing a single unnamed column.

3.1.5.40.29 App Principal Information Result Set

If the @AppPrincipalName parameter is not NULL, the App Principal Info Result Set MUST be returned as defined in section 2.2.4.29.

3.1.5.40.30 App Principal Permissions Result Set
The **App Principal Permissions Result Set** is returned to indicate what permissions the specified **app principal** has for the **site (2)** or **list (1)** as specified in section 2.2.4.30. If the specified app principal has permissions for the specified site (2) or list (1), the **App Principal Permissions Result Set** MUST be returned.

### 3.1.5.41 proc_GetUniqueScopesInList

The **proc_GetUniqueScopesInList** stored procedure is invoked to get the **WSS ACL Format** (section 2.2.3.6) information for all the unique security scopes of folders and list items contained in a specified list.

```sql
PROCEDURE proc_GetUniqueScopesInList(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @ListId                        uniqueidentifier,
    @GetAll                        bit,
    @ThresholdRowCount             int,
    @RequestGuid                   uniqueidentifier   =NULL      OUTPUT
);
```

**@SiteId**: The **Site Collection Identifier** of the site collection containing the list.

**@WebId**: The **Site Identifier** of the **site (2)** that contains the list.

**@ListId**: The **List Identifier** of the list for which the **WSS ACL Format** information is requested.

**@GetAll**: A bit specifying whether to fetch **WSS ACL Format** information for all unique security scopes. If this bit is set to "0", **WSS ACL Format** information for most @ThresholdRowCount security scopes MUST be returned in the **Unique Permissions Result Set** (section 2.2.4.26), else **WSS ACL Format** information for all unique security scopes MUST be returned in the **Unique Permissions Result Set**.

**@ThresholdRowCount**: This parameter specifies the number of security scopes to return when @GetAll is set to "0". The value MUST NOT be 0.

**@RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_GetUniqueScopesInList** stored procedure returns an integer return code which MUST be 0.

The **proc_GetUniqueScopesInList** stored procedure MUST return a single result set.

### 3.1.5.41.1 Unique Permissions Result Set

The **Unique Permissions Result Set** returns the **WSS ACL Format** (section 2.2.3.6) information for all the unique security scopes of folders and list items contained in a specified list. The **Unique Permissions Result Set** MUST be returned if the list has 1 or more folders or list items with unique security scopes. The **Unique Permissions Result Set** MUST return one row for every folder or list item with a unique security scope. If the @GetAll bit is set to "0", then this result set MUST return at most @ThresholdRowCount rows.

The **Unique Permissions Result Set** is defined the Common Result Sets **Unique Permissions Result Set** (section 2.2.4.26).

### 3.1.5.41.2 NULL Unique Permissions Result Set

The **NULL Unique Permissions Result Set** MUST return one row if the list has no folders or list items with unique security scopes. The **NULL Unique Permissions Result Set** is defined in the Common Result Sets **NULL Unique Permissions Result Set** (section 2.2.4.17).
3.1.5.42 proc_GetVersion

The proc_GetVersion stored procedure is invoked to get the component version number string associated with the specified version identifier GUID (using the format major.minor.phase.build; for example, 3.0.106.0).

```sql
PROCEDURE proc_GetVersion(
    @VersionId uniqueidentifier,
    @Version nvarchar(64) OUTPUT
);
```

@VersionId: The version identifier of the component version number to retrieve.

@Version: An output parameter containing the component version number string that matches the specified version identifier. This value MUST be unchanged from the input value if the specified @VersionId value does not have a matching component version number.

Return Values: The proc_GetVersion stored procedure MUST return value 0.

The front-end Web server can define one or more version numbers. Each of the version numbers is identified by a version identifier. When a database is created by the front-end Web server on the back-end database server, the front-end Web server stores the version numbers with different version identifiers in the Versions table (section 2.2.6.12). When the front-end Web server connects to a back-end database server, the front-end Web server SHOULD retrieve the version identifiers using stored procedure proc_GetVersion and make sure the versions are within an acceptable range defined by the front-end Web server. Otherwise, the front-end Web server MUST NOT connect to the back-end database server.

The version identifiers and the acceptable version numbers for this protocol that are used when communicating with a content database are listed in the following table.

<table>
<thead>
<tr>
<th>Version Id GUID</th>
<th>Acceptable version range</th>
</tr>
</thead>
<tbody>
<tr>
<td>'00000000-0000-0000-0000-000000000000'</td>
<td>14.0.4006.1010 - 14.0.4006.9999</td>
</tr>
<tr>
<td>'6333368D-85F0-4EF5-8241-5252B12B2E50'</td>
<td>4.0.116.0 - 4.0.116.0</td>
</tr>
<tr>
<td>'1A707EF5-45B2-4235-9327-021E5F9B8BB0'</td>
<td>4.0.6.0 - 4.0.6.0</td>
</tr>
<tr>
<td>'25EB5CEE-15BD-4954-BD4E-2624D5878D8C'</td>
<td>14.0.4006.1010 - 14.0.4006.9999</td>
</tr>
</tbody>
</table>

When the protocol is used to communicate with a configuration database, the list of version identifiers used are the following.

<table>
<thead>
<tr>
<th>Version Id GUID</th>
<th>Acceptable version range</th>
</tr>
</thead>
<tbody>
<tr>
<td>'00000000-0000-0000-0000-000000000000'</td>
<td>14.0.4006.1010 - 14.0.4006.9999</td>
</tr>
<tr>
<td>'F4D348C4-A6E9-4ed5-BDB2-2358B74EF902'</td>
<td>4.0.116.0 - 4.0.116.0</td>
</tr>
<tr>
<td>'60B1F2BE-5130-45AB-AF1D-EDD34E626B5D'</td>
<td>4.0.6.0 - 4.0.6.0</td>
</tr>
</tbody>
</table>

The acceptable version numbers specified in the preceding tables can change when the front-end Web server is updated. As a result, the updated front-end Web server might not be able to communicate with the back-end database server. To re-enable front-end Web server to back-end database server communication, the front-end Web server MUST use an upgrade process to modify any data structure on the back end database server to accommodate any changes that might have occurred to this protocol, and to update the version numbers to match the new acceptable version numbers. Upgrade logic is implementation-specific.
The **proc_GetVersion** stored procedure MUST NOT return any **result sets**.

### 3.1.5.43 proc_GetWebMetaInfo

The **proc_GetWebMetaInfo** stored procedure is invoked to request metadata information for a site (2).

```sql
PROCEDURE proc_GetWebMetaInfo(
    @WebSiteId                     uniqueidentifier,
    @WebDirName                    nvarchar(256),
    @WebLeafName                   nvarchar(128),
    @DGCacheVersion                bigint,
    @SystemId                      varbinary(512) = NULL,
    @AppPrincipalName              nvarchar(256),
    @IsHostHeaderAppPrincipalName  bit,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
)
```

@WebSiteId: The **site collection identifier** for the **site collection** containing the site (2) whose metadata is requested.

@WebDirName: The **directory name** part of the site location.

@WebLeafName: The **leaf name** part of the site location.

@DGCacheVersion: The version number of the **domain group** map cache in the **front-end Web server**. This can be compared with the domain group map cache version number on the **back-end database server** returned in the **Domain Group Cache Versions Result Set** (section 3.1.5.43.2) to determine whether updates are needed. If the @DGCacheVersion parameter is set to -2, then the information about the domain group map cache is not requested, and the **Domain Group Cache BEDS Update Result Set** (section 3.1.5.43.3) and the **Domain Group Cache WFE Update Result Set** (section 3.1.5.43.4) MUST NOT be returned.

@SystemId: The **SystemID** of the current user requesting the site metadata information.

@AppPrincipalName: The **app principal** identifier or the **application web domain Id** associated with the current user. If the current user is not associated with an application principal, this parameter MUST be NULL.

@IsHostHeaderAppPrincipalName: If this parameter is set to "1", the @AppPrincipalName parameter MUST be the app principal identifier; otherwise, it MUST be the application web domain Id.

@RequestGuid: The optional **request identifier** for the current request.

**Return Values:** This stored procedure returns an integer **return code**, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>A site (2) was not found at the specified location.</td>
</tr>
<tr>
<td>1271</td>
<td>The site collection is locked for read and write operations.</td>
</tr>
</tbody>
</table>

**Result Sets:** This stored procedure MUST return **result sets** as described in the following sections: **Site MetaInfo Result Set** (section 3.1.5.43.1), **Domain Group Cache Versions Result Set** (section 3.1.5.43.2), **Domain Group Cache BEDS Update Result Set** (section 3.1.5.43.3), **Domain Group Cache WFE Update Result Set** (section 3.1.5.43.4), **Site Metadata Result Set** (section 3.1.5.43.5), and **Event Receivers Result Set** (section 3.1.5.43.6).
3.1.5.43.1  Site MetaInfo Result Set

The Site MetaInfo Result Set MUST be returned. The Site MetaInfo Result Set MUST contain a single row containing the metainfo for a valid site. If the specified site (2) is not found, zero rows MUST be returned and the stored procedure MUST NOT return any more result sets. The Site MetaInfo Result Set is defined in section Site MetaInfo Result Set (section 2.2.4.25), located in the Common Result Sets (section 2.2.4).

3.1.5.43.2  Domain Group Cache Versions Result Set

The Domain Group Cache Versions Result Set contains information about the version numbers associated with the domain group map caches on the front-end Web server and on the back-end database server for the specified site collection.

The Domain Group Cache Versions Result Set MUST be returned if the specified site (2) is found, and MUST contain one row of version number data. If the value specified by the @DGCacheVersion parameter is set to -2, then all columns returned MUST have the value -2, indicating that the value MUST NOT be used for comparison. The Domain Group Cache Versions Result Set is defined in section Domain Group Cache Versions Result Set (section 2.2.4.4), located in the Common Result Sets (section 2.2.4).

3.1.5.43.3  Domain Group Cache BEDS Update Result Set

The Domain Group Cache BEDS Update Result Set contains information to be used in recomputing the domain group map cache.

The Domain Group Cache BEDS Update Result Set returns only if the Domain Group Cache Versions Result Set (section 3.1.5.43.2) is returned and the @DGCacheVersion parameter is not set to -2 and the real domain group version is more recent than the cached version. (The value of RealVersion is greater than the value of CachedVersion in the Domain Group Cache Versions Result Set).

If the Domain Group Cache BEDS Update Result Set is returned, then it indicates that the database's copy of the domain group map cache is out of date.

When returned, the Domain Group Cache BEDS Update Result Set MUST contain one row for each domain group that is a member of a permission level in the site collection, ordered by the User Identifier (section 2.2.1.1.13) of the domain groups.

This result set is defined in section Domain Group Cache BEDS Update Result Set (section 2.2.4.3), located in the Common Result Sets (section 2.2.4).

3.1.5.43.4  Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the domain group map cache.

The Domain Group Cache WFE Update Result Set returns only if the Domain Group Cache Versions Result Set is returned and the @DGCacheVersion parameter is not set to -2 and the cached version is up to date (the value of RealVersion is not greater than the value of CachedVersion in the Domain Group Cache Versions Result Set).

If the Domain Group Cache WFE Update Result Set is returned, one row MUST be returned. The Domain Group Cache WFE Update Result Set is defined in section Domain Group Cache WFE Update Result Set (section 2.2.4.5), located in the Common Result Sets (section 2.2.4).

3.1.5.43.5  Site Metadata Result Set

The Site Metadata Result Set contains metadata for the specified site (2).
The **Site Metadata Result Set** MUST be returned if the specified site (2) is found, and MUST contain a single row corresponding to the specified site. If the site (2) is locked from both read and write operations, the **stored procedure** MUST NOT return any more **result sets**.

The **Site Metadata Result Set** is defined in section **Site Metadata Result Set** (section 2.2.4.24), located in the Common Result Sets (section 2.2.4).

### 3.1.5.43.6 Event Receivers Result Set

The **Event Receivers Result Set** contains information about the **event receivers** defined for the specified site. There MUST be one row in the **Event Receivers Result Set** for each event receiver that is registered for this site (2). The **Event Receivers Result Set** MUST be returned on successful execution.

The **Event Receivers Result Set** is defined in section **Event Receivers Result Set** (section 2.2.4.11), located in the Common Result Sets (section 2.2.4).

### 3.1.5.44 proc_GetWebMetainfoByUrl

The **proc_GetWebMetainfoByUrl** stored procedure is invoked to request site metadata information for the site (2) containing a specified Uniform Resource Locator (URL).

```sql
PROCEDURE proc_GetWebMetainfoByUrl(
  @WebSiteId uniqueidentifier,
  @Url nvarchar(260),
  @DGCacheVersion bigint,
  @SystemId varbinary(512) = NULL,
  @AppPrincipalName nvarchar(256),
  @IsHostHeaderAppPrincipalName bit,
  @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@**WebSiteId**: The **Site Collection Identifier** (section 2.2.1.1.9) for the **site collection** containing the site (2) or document whose metadata is requested.

@**Url**: A URL in store-relative form for the site (2) or for a document within the site (2).

@**DGCacheVersion**: The version number of the Domain Group Map Cache in the **front-end Web server**. This can be compared with the Domain Group Map Cache version number on the **back-end database server** returned in the **Domain Group Cache Versions Result Set** (section 3.1.5.44.3) to determine whether updates are needed. If the @**DGCacheVersion** parameter is set to "-2", then information about the Domain Group Map Cache is not requested, and the **Domain Group Cache BEDS Update Result Set** (section 3.1.5.44.4) and the **Domain Group Cache WFE Update Result Set** (section 3.1.5.44.5) MUST NOT be returned.

@**SystemId**: The **SystemID** of the current user requesting the site metadata information.

@**AppPrincipalName**: The **app principal** identifier or the **application web domain Id** associated with the current user. If the current user is not associated with an application principal, this parameter MUST be NULL.

@**IsHostHeaderAppPrincipalName**: If this parameter is set to "1", the @**AppPrincipalName** parameter MUST be the app principal identifier; otherwise, it MUST be the application web domain Id.

@**RequestGuid**: The optional **request identifier** for the current request.

**Return Values**: This stored procedure returns an integer return code, which MUST be listed in the following table.
This stored procedure MUST return 0, 5, or 6 **result sets**. Some of the result sets are returned conditionally, and all result sets returned will be sent in the order specified in the following sections.

### 3.1.5.44.1 Site URL Result Set

The **Site URL Result Set** MUST be returned if the values for the @Url and @WebSiteId parameters are valid. Otherwise, the stored procedure MUST NOT return any **result sets**. The **Site URL Result Set** MUST contain a single row. The **T-SQL** syntax for the result set is as follows:

```
{WebUrl}    nvarchar(260),
{AppWebDomainId} varchar(8);
```

**{WebUrl}**: The **URL** in store-relative form of the **site (2)** that contains the URL specified in the @Url parameter.

**{AppWebDomainId}**: The **application web domain Id** associated with the site (2) that contains the URL specified in the @Url parameter. This MAY be NULL.

### 3.1.5.44.2 Site MetaInfo Result Set

The **Site MetaInfo Result Set** MUST contain a single row containing the metainfo for the **site (2)**. For details about the **Site MetaInfo Result Set**, see section 2.2.4.25.

### 3.1.5.44.3 Domain Group Cache Versions Result Set

The **Domain Group Cache Versions Result Set** contains information about the version numbers associated with the domain group map caches on the **front-end Web server** and on the **back-end database server** for the specified **site collection**.

The **Domain Group Cache Versions Result Set** MUST contain one row of version number data. If the value specified for the @DGCacheVersion parameter is set to "-2", then all columns returned MUST have the value "-2", indicating that the value MUST NOT be used for comparison. The **Domain Group Cache Versions Result Set** is defined in section 2.2.4.4.

### 3.1.5.44.4 Domain Group Cache BEDS Update Result Set

The **Domain Group Cache BEDS Update Result Set** contains information to be used in recomputing the domain group map cache, which contains the mapping of **domain groups** to the site permission levels of which they are members. The presence of the **Domain Group Cache BEDS Update Result Set** means the database's copy of the domain group map cache is out of date.

The **Domain Group Cache BEDS Update Result Set** MUST be returned if the @DGCacheVersion parameter does not equal "-2" and the value of RealVersion is greater than the value of CachedVersion in the results of the **Domain Group Cache Versions Result Set**. Otherwise, this **result set** MUST NOT be returned.

If the **Domain Group Cache BEDS Update Result Set** is returned, it MUST contain one row for each domain group that is a member of a site permission level in the **site collection**, ordered by the **User Identifier** (section 2.2.1.1.13) of the domain groups.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>1168</td>
<td>The site collection does not exist.</td>
</tr>
<tr>
<td>1271</td>
<td>The site (2) is locked for read and write operations.</td>
</tr>
</tbody>
</table>
This result set is defined in section 2.2.4.3.

3.1.5.44.5  Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the domain group map cache. If the Domain Group Cache WFE Update Result Set is returned, then it indicates that the back-end database server Domain Group map cache is up to date, and the front-end Web server cache can be refreshed if necessary.

The Domain Group Cache WFE Update Result Set MUST be returned on successful determination of the site (2) if the @DGCacheVersion parameter is not set to "-2" and the value of RealVersion is less than or equal to the value of CachedVersion in the results of the Domain Group Cache Versions Result Set. Otherwise, the Domain Group Cache WFE Update Result Set MUST NOT be returned.

If the Domain Group Cache WFE Update Result Set is returned, it MUST contain one row.

The Domain Group Cache WFE Update Result Set is defined in section 2.2.4.5.

3.1.5.44.6  Site Metadata Result Set

The Site Metadata Result Set contains specialized site metadata. If the site (2) is locked for read and write operations, the stored procedure MUST NOT return any more result sets. There MUST be one row in the Site Metadata Result Set.

The Site Metadata Result Set is defined in section 2.2.4.24.

3.1.5.44.7  Event Receivers Result Set

The Event Receivers Result Set contains information about the event receivers defined for this site (2). There MUST be one row in the Event Receivers Result Set for each event receiver registered for this site (2). The Event Receivers Result Set MUST be returned on successful completion.

The Event Receivers Result Set is defined in section 2.2.4.11.

3.1.5.45  proc_ListDocumentVersions

The proc_ListDocumentVersions stored procedure is invoked to list all available version history information for a specified document.

PROCEDURE proc_ListDocumentVersions(
    @DocSiteId uniqueidentifier,
    @DocWebId uniqueidentifier,
    @DocDirName nvarchar(256),
    @DocLeafName nvarchar(128),
    @MaxLevel tinyint,
    @UserId int,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

@DocWebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the document.

@DocDirName: The directory name containing the document.

@DocLeafName: The leaf name containing the document.
@MaxLevel: A Publishing Level Type (section 2.2.2.6) indicating the maximum publishing level value of the document that SHOULD be returned to the front-end Web server in the Document Versions Result Set (section 3.1.5.45.3) if multiple publishing levels of the document are available. See the @UserId parameter for an exception to this.

@UserId: This parameter is the User Identifier (section 2.2.1.1.13) of the current user. If the current user is the owner of one or more versions of the document at any publishing level, the version information MUST be returned in the Document Versions Result Set, ignoring the value specified in the @MaxLevel parameter.

@RequestGuid: The optional request identifier for the current request.

Return Values: This stored procedure returns an integer return code, which MUST be included in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>2</td>
<td>The document specified is not a file (that is, it does not have a Document Store Type (section 2.2.2) of 0).</td>
</tr>
</tbody>
</table>

This stored procedure MUST return two or three result sets upon successful completion.

### 3.1.5.45.1 Individual URL Security Result Set

The Individual URL Security Result Set contains security information about the specified document. If the document does not exist, but the specified document URL is within a list or document library, security information is returned from the document's parent object, such as the list or document library within which it would be contained within.

The Individual URL Security Result Set MUST only be returned if the document URL is contained within a list or document library. Otherwise, the NULL Individual URL Security Result Set (section 3.1.5.45.2) MUST be returned instead. If returned, the Individual URL Security Result Set MUST contain a single row.

The Individual URL Security Result Set is defined in section Individual URL Security Result Set (section 2.2.4.12), located in the Common Result Sets (section 2.2.4).

### 3.1.5.45.2 NULL Individual URL Security Result Set

The NULL Individual URL Security Result Set indicates that the specified document URL is not contained within a list or document library. The NULL Individual URL Security Result Set MUST only be returned if the document URL is not contained within a list or document library.

The NULL Individual URL Security Result Set MUST return a single row and is defined in section NULL Individual URL Security Result Set (section 2.2.4.16), located in the Common Result Sets (section 2.2.4).

### 3.1.5.45.3 Document Versions Result Set

The Document Versions Result Set returns version information for a specified document. The Document Versions Result Set MUST only be returned when the specified document has a Document Store Type (section 2.2.2.4) of 0. The Document Versions Result Set MUST contain document version information corresponding to publishing levels less than or equal to the specified publishing level, or corresponding to any publishing levels owned by the current user.
Size                           int,
CheckinComment                 nvarchar(1023),
MetaInfo                       varbinary(max),
{IsTip}                        bit,
Level                          tinyint,
DraftOwnerId                   int,
StreamSchema                   tinyint,
{NextBSN}                      bigint;

**TimeCreated:** A timestamp in UTC format specifying when this document was created.

**VersionNumber:** A counter incremented any time a change is made to this document, used for internal conflict detection.

**Size:** The number of bytes in the document stream.

**CheckinComment:** An optional user-provided description used when a document is being checked in or published. This value MUST be NULL for documents that have not been checked in or published.

**MetaInfo:** A metadict for the document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11.

**{IsTip}:** If set to 1, the document is a current version, otherwise it is a historical version.

**Level:** A Publishing Level Type (section 2.2.2.6) value specifying the publishing level of the document.

**DraftOwnerId:** The user that published this document as a draft. This value is only non-NULL if the document is a draft version. See Publishing Level Type for more information regarding the different publishing levels.

**StreamSchema:** The current stream schema of the document being returned.

**{NextBSN}:** The current BLOB sequence number of the document being returned.

### 3.1.5.46   proc_ListRbsStores

The proc_ListRbsStores stored procedure is invoked to enumerate the remote BLOB storage stores with which the back-end database server has been configured, if any.

```
PROCEDURE proc_ListRbsStores (  
    @RequestGuid uniqueidentifier = null OUTPUT,
);
```

@RequestGuid: The optional request identifier for the current request.

Return values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Default return value</td>
</tr>
</tbody>
</table>

Result Sets:

This procedure MUST return the List RBS Stores Result Set.

### 3.1.5.46.1  List RBS Stores Result Set
blob_store_name: A remote BLOB storage store name with which the BEDS has been configured.

3.1.5.47 proc_ListUrls

The proc_ListUrls stored procedure retrieves metadata for a document specified by a URL and the documents contained within it, if any.

PROCEDURE proc_ListUrls(
    @DirSiteId uniqueidentifier,
    @DirWebId uniqueidentifier,
    @DirFullUrl nvarchar(260),
    @AttachmentsFlag tinyint,
    @ClientTimeStam datetime,
    @FetchLinkInfo bit,
    @IncludeThicketDirs bit,
    @IncludeListItems bit,
    @UserId int,
    @ItemLimit int,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);

@DirSiteId: The Site Collection Identifier (section 2.2.1.1.9) of the site collection containing the document specified by a URL.

@DirWebId: The Site Identifier (section 2.2.1.1.11) of the site (2) containing the document.

@DirFullUrl: The URL in store-relative form specifying the document.

@AttachmentsFlag: An Attachments Flag (section 2.2.1.2.1) value specifying whether the document is, or is contained within, a folder for attachments.

@ClientTimeStam: A datetime that specifies a limiting time on the data in the result sets returned. See the description of the result sets in the following sections for the specific effects of this parameter value.

@FetchLinkInfo: A bit flag specifying whether to return the Link Info Result Set (section 3.1.5.47.6). If this parameter is set to 1 and the specified document is a folder, the Link Info Result Set MUST be returned.

@IncludeThicketDirs: A bit flag specifying whether to return thicket folders in the Contained Document Metadata Result Set (section 3.1.5.47.7). If this parameter is set to 1 and the specified document is a folder, any thicket folders MUST be included in the Contained Document Metadata Result Set.

@IncludeListItems: A bit flag specifying whether to include list items which are files in the Link Info Result Set and the Contained Document Metadata Result Set. If this parameter is set to 1 and the document is a folder, list items with a Document Store Type (section 2.2.2.4) of 0 MUST be included in the Link Info Result Set and the Contained Document Metadata Result Set.

@UserId: The User Identifier (section 2.2.1.1.13) for the current user.

@ItemLimit: This parameter MUST be ignored if set to a value that is less than or equal to 0. Otherwise, if this parameter is greater than 0 and the document specified by the @DirFullUrl parameter contains a sum total of list items and folders that exceed the value specified by this parameter, the stored procedure MUST fail execution. See the below Return Values for more information.
@RequestGuid: The optional request identifier for the current request.

Return Values: This stored procedure returns an integer return code that MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The document URL is not valid, or the document is not a folder or site (2).</td>
</tr>
<tr>
<td>36</td>
<td>The stored procedure failed to complete successfully. This value is returned if the @ItemLimit parameter is greater than 0, this list (1) is not exempt from resource throttling operations, and the number of list items combined with the number of folders in the list (1) exceeds the @ItemLimit parameter.</td>
</tr>
</tbody>
</table>

This stored procedure MUST return two to six result sets, as described in the following sections, in the following order.

3.1.5.47.1 Individual URL Security Result Set

The Individual URL Security Result Set contains security information about the specified document. If the document does not exist, but the specified document URL is within a list (1) or document library, security information is returned from the document's parent object, such as the list (1) or document library within which it would be contained.

The Individual URL Security Result Set MUST only be returned if the document URL is contained within a list (1) or document library. Otherwise, the NULL Individual URL Security Result Set MUST be returned instead. If returned, the Individual URL Security Result Set MUST contain a single row.

The result set is defined in Individual URL Security Result Set (section 2.2.4.12), located in the Common Result Sets (section 2.2.4).

3.1.5.47.2 NULL Individual URL Security Result Set

The NULL Individual URL Security Result Set indicates that the specified document URL is not contained within a list (1) or document library. The NULL Individual URL Security Result Set MUST only be returned if the document URL is not contained within a list (1) or document library.

The result set MUST return a single row and is defined in NULL Individual URL Security Result Set (section 2.2.4.16), located in the Common Result Sets (section 2.2.4).

3.1.5.47.3 Server Time Result Set

The Server Time Result Set returns the current time from the back-end database server in UTC. The Server Time Result Set MUST be returned and MUST contain a single row of data.

The result set is defined in Server Time Result Set (section 2.2.4.20), located in the Common Result Sets (section 2.2.4).

3.1.5.47.4 Subsite List Result Set

The Subsite List Result Set contains an unordered list of URLs in store-relative form for all subsites whose parent site is the site (2) specified by the @DirWebId parameter.

If the specified document is not a folder, the Subsite List Result Set MUST NOT be returned. Otherwise the Subsite List Result Set MUST be returned and MUST contain one row for each subsite within the specified site (2), and it MUST contain no rows if there are no such subsites.
The result set is defined in **URL Result Set** (section 2.2.4.27), located in the Common Result Sets (section 2.2.4).

### 3.1.5.47.5 Document Metadata Result Set

The **Document Metadata Result Set** contains the metadata for the specified document. If the document is not a folder or if the @DirFullUrl parameter contains an empty string, the Document Metadata Result Set MUST NOT be returned. Otherwise, the Document Metadata Result Set MUST contain one row with the metadata information for the document.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>{FullUrl}</td>
<td>nvarchar(260),</td>
</tr>
<tr>
<td>Type</td>
<td>tinyint,</td>
</tr>
<tr>
<td>MetaInfoTimeLastModified</td>
<td>datetime,</td>
</tr>
<tr>
<td>MetaInfo</td>
<td>varbinary(max),</td>
</tr>
<tr>
<td>Size</td>
<td>int,</td>
</tr>
<tr>
<td>TimeCreated</td>
<td>datetime,</td>
</tr>
<tr>
<td>TimeLastModified</td>
<td>datetime,</td>
</tr>
<tr>
<td>ETagVersion</td>
<td>int,</td>
</tr>
<tr>
<td>DocFlags</td>
<td>int,</td>
</tr>
<tr>
<td>[ListType]</td>
<td>nvarchar(38),</td>
</tr>
<tr>
<td>tp_Name</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>CacheParseId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>(GhostDirName)</td>
<td>nvarchar(256),</td>
</tr>
<tr>
<td>(GhostLeafName)</td>
<td>nvarchar(128),</td>
</tr>
<tr>
<td>[tp_Login]</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>(CheckoutDate)</td>
<td>datetime,</td>
</tr>
<tr>
<td>(CheckoutExpires)</td>
<td>datetime,</td>
</tr>
<tr>
<td>VirusStatus</td>
<td>int,</td>
</tr>
<tr>
<td>VirusInfo</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>SetupPathVersion</td>
<td>tinyint,</td>
</tr>
<tr>
<td>SetupPath</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>SetupPathUser</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>NextToLastTimeModified</td>
<td>datetime,</td>
</tr>
<tr>
<td>UIVersion</td>
<td>int,</td>
</tr>
<tr>
<td>CheckinComment</td>
<td>nvarchar(1023),</td>
</tr>
<tr>
<td>WelcomePageUrl</td>
<td>nvarchar(260),</td>
</tr>
<tr>
<td>WelcomePageParameters</td>
<td>nvarchar(max),</td>
</tr>
<tr>
<td>tp_Flags</td>
<td>bigint,</td>
</tr>
<tr>
<td>AoI</td>
<td>varbinary(max),</td>
</tr>
<tr>
<td>AnonymousPermMask</td>
<td>bigint,</td>
</tr>
<tr>
<td>DraftOwnerId</td>
<td>int,</td>
</tr>
<tr>
<td>Level</td>
<td>tinyint,</td>
</tr>
<tr>
<td>ParentVersion</td>
<td>int,</td>
</tr>
<tr>
<td>TransformerId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>ParentLeafName</td>
<td>nvarchar(128),</td>
</tr>
<tr>
<td>ProgId</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>DoclibRowId</td>
<td>int,</td>
</tr>
<tr>
<td>tp_DefaultWorkflowId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>ListId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>ItemChildCount</td>
<td>int,</td>
</tr>
<tr>
<td>FolderChildCount</td>
<td>int,</td>
</tr>
<tr>
<td>MetaInfoVersion</td>
<td>int,</td>
</tr>
<tr>
<td>{CurVerMetaInfo}</td>
<td>varbinary(max),</td>
</tr>
<tr>
<td>ContentVersion</td>
<td>int,</td>
</tr>
<tr>
<td>IsDirty</td>
<td>bit,</td>
</tr>
<tr>
<td>{NextBSN}</td>
<td>bigint,</td>
</tr>
<tr>
<td>{StreamSchema}</td>
<td>tinyint,</td>
</tr>
<tr>
<td>{InternalVersion}</td>
<td>int</td>
</tr>
</tbody>
</table>

**Id:** The **Document Identifier** (section 2.2.1.1.2) of the document.

**{FullUrl}:** The **URL** in **store-relative form** for the document.
**Type:** The **Document Store Type** (section 2.2.4) of the document.

**MetaInfoTimeLastModified:** A datetime with a timestamp in UTC format specifying the last time the **MetaInfo** column value of the document was changed. This value MUST be NULL if the **MetaInfo** column value of the document has never been changed.

**MetaInfo:** A *metadict* for the document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value MUST be NULL if the document does not have any metadict associated with it.

**Size:** The number of bytes in the **document stream** of the document. This value MUST be NULL if the document does not have a document stream.

**TimeCreated:** A datetime with a timestamp in UTC format specifying when the document was created.

**TimeLastModified:** A datetime with a timestamp in UTC format specifying when the document was last modified.

**ETagVersion:** A counter used for internal conflict detection that is incremented any time a change is made to the document.

**DocFlags:** A **Doc Flags** value (section 2.2.3) describing the document. This value MUST be NULL if the document does not have any **Doc Flags** associated with it.

**{ListType}:** A packed combination of the **List Base Type** (section 2.2.1.2.11) and **List Server Template** (section 2.2.1.2.12) values of the list containing this document, consisting of the **List Server Template** value multiplied by 256 and added to the value of the **List Base Type**.

**tp_Name:** The **List Identifier** (section 2.2.1.5) of the list (1) containing this document.

**{ListTitle}:** If the document URL is the **root folder** of a list (1), this contains the **display name** of the list (1). Otherwise, this value MUST be NULL.

**CacheParseId:** This value MUST be NULL.

**{GhostDirName}:** This value MUST be NULL.

**{GhostLeafName}:** This value MUST be NULL.

**{tp_Login}:** This value MUST be NULL.

**{CheckoutDate}:** This value MUST be NULL.

**{CheckoutExpires}:** This value MUST be NULL.

**VirusStatus:** A **Virus Status** (section 2.2.1.2.17) value specifying the current virus state of the document. This value can be NULL if the document has not been processed by a virus scanner.

**VirusInfo:** A string containing a provider-specific message returned by the virus scanner when it last processed the document. This value can be NULL if the document has not been processed by a virus scanner.

**SetupPathVersion:** If this is an **uncustomized** document, this specifies the setup path location that the **SetupPath** fragment is relative to. This value MUST NOT be NULL. The following are valid values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>The <strong>SetupPath</strong> is relative to the install location of Windows SharePoint Services 2.0 on the <strong>front-end Web server</strong> (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>3</td>
<td>The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).</td>
</tr>
<tr>
<td>4</td>
<td>The SetupPath is relative to the install location of SharePoint Foundation 2010 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).</td>
</tr>
<tr>
<td>15</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Foundation 2013 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\15).</td>
</tr>
<tr>
<td>16</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Server 2016 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\16).</td>
</tr>
</tbody>
</table>

**SetupPath:** If the document is now or once was uncustomized, this contains the setup path fragment relative to the base setup path described above by the SetupPathVersion value where the document stream of the document can be found. This value can be NULL.

**SetupPathUser:** If the document is now or once was uncustomized, this contains the login name of the user who created the uncustomized document. This value can be NULL.

**NextToLastTimeModified:** The value of TimeLastModified from the previous time when the document was last saved. If the NextToLastTimeModified value matches the TimeLastModified value when the change occurred and the client has a version of the document that it has successfully modified, the client can safely submit the document to the server despite what appears to be an intervening edit to the document. This value MUST be NULL if the document has never been saved.

**UIVersion:** The UI version number for the document.

**CheckinComment:** An optional user-supplied description provided when the document is checked in or published. This value can be NULL.

**WelcomePageUrl:** If the document is a folder, this specifies an optional page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself, and MUST be contained within that folder. Attempts to break out of the folder, such as "/../somepage.aspx", are not valid. This value can be NULL.

**WelcomePageParameters:** Contains optional URL parameters to specify as part of the WelcomePageUrl value. These parameters start at either the query string signifier '?' or the bookmark signifier '#'. This value can be NULL.

**tp_Flags:** The List Flags value (section 2.2.2.5) for the list (1) that contains the document.

**Acl:** The binary serialization of the WSS ACL Format (section 2.2.3.6) for the document. The WSS ACL is either explicitly defined for the document or inherited from the parent object of the document. This value can be NULL.

**AnonymousPermMask:** A WSS Rights Mask (section 2.2.2.15) that indicates the permissions granted anonymous users or those users that have no specific permissions to the document. This value can be NULL if anonymous access to the document is not allowed.

**DraftOwnerId:** The User Identifier (section 2.2.1.1.13) of the user that published the document as a draft. This value MUST be NULL if the document is not a draft version.

**Level:** A Publishing Level Type value (section 2.2.2.6) specifying the publishing status of the document.
**ParentVersion**: If the document is a transformed version of another document, this is the UI version value from the original document. This value MUST be NULL if the document is not the product of a document transformation.

**TransformerId**: If the document is a transformed version of another document, this is the **GUID** of the agent that performed the transformation. This value MUST be NULL if the document is not the product of a document transformation.

**ParentLeafName**: If the document is a transformed version of another document, this is the leaf name of the original document. The original document MUST be in the same folder as the transformed document. If either is moved, the relationship is broken. This value MUST be NULL if the document is not the product of a document transformation.

**ProgId**: Specifies a preferred application to open the document. The **ProgId** value is used to distinguish between different applications that save files with a given **file extension** (for example, different editors for .HTML or .XML files). This value MUST be NULL if a **ProgId** was not specified when the document was saved.

**DoclibRowId**: The identifier for a row in a **document library** for the document. If the document is not contained in a list (1), this value MUST be NULL.

**tp_DefaultWorkflowId**: The **Workflow Identifier** (section 2.2.1.1.16) corresponding to the workflow to be invoked if the document is in a moderated list (1) and the document is submitted for approval as part of a check in.

**ListId**: The list identifier of the list (1) that contains the document. If the document is not contained in a list (1), this value MUST be NULL.

**ItemChildCount**: The value MUST be the number of list items for the list (1) that contains the document.

**FolderChildCount**: The value MUST be the number of folders that exist in the list (1) that contains the document.

**MetaInfoVersion**: A counter used for internal conflict detection that is incremented any time a change is made to the **MetaInfo** for this document.

**{CurVerMetaInfo}**: The value MUST be NULL.

**ContentVersion**: The version number of the document stream being returned.

**IsDirty**: A bit that specifies this document MUST have implementation-specific processing performed before its stream can be returned. If this document does not require processing, this value MUST be zero.

**{NextBSN}**: The current **BLOB** sequence number of the document being returned.

**{StreamSchema}**: The current stream schema of the document being returned.

**{InternalVersion}**: The internal version number of the document being returned.

### 3.1.5.47.6 Link Info Result Set

The **Link Info Result Set** returns information about all forward links from and backward links to the documents contained within the specified document. If the specified document is not a folder, or the **@FetchLinkInfo** parameter is not set to 1, then the **Link Info Result Set** MUST NOT be returned. Otherwise, if the document is a folder and the **@FetchLinkInfo** parameter is set to 1, then the **Link Info Result Set** MUST be returned. The **Link Info Result Set** MUST contain one row for each link that has been modified after the value in the **@ClientTimeStamp** parameter for each contained document in the site collection which has its directory name equal to the value of the **@DirFullUrl**.
parameter, and the contained document has a **Document Store Type** (section 2.2.4.4) of 0 and the @IncludeListItems parameter is set to 1, the document is in a document library, or the document can have a document stream, and for forward links, is a published or draft version which is not checked out to the specified user, or is a checked out version which is checked out to the user.

The **Link Info Result Set** is defined using T-SQL syntax as described in section 2.2.4.21, located in the Common Result Sets (section 2.2.4).

### 3.1.5.47.7 Contained Document Metadata Result Set

The **Contained Document Metadata Result Set** contains the metadata information for the documents contained within the specified document. If the specified document is not a folder, the Contained Document Metadata Result Set MUST NOT be returned. Otherwise, the Contained Document Metadata Result Set MUST return one row for each document that matches the following conditions:

- It belongs to **site collection** whose directory name is equal to the value of the @DirFullUrl parameter.
- Either:
  - It is a published or draft version which is not checked out to the specified user.
  - Or, it is a checked out version which is checked out to the user.
- Either:
  - The contained document has a **Document Store Type** (section 2.2.4.4) of zero and the @IncludeListItems parameter is set to one, or the document is in a document library, or the document can have a document stream.
  - Or, the contained document is a folder, and the folder is not a thicket supporting file, and either the folder is not a thicket folder or the @IncludeThicketDirs parameter is set to one.
- Either:
  - The **app web domain identifier** of the site (2) containing the document is equal to the app web domain identifier of the site (2) whose identifier is equal to @DirWebId parameter.
  - Or, both site (2) containing the document and the site (2) whose identifier is equal to @DirWebId parameter, don't have app web domain identifier.

```sql
Id               uniqueidentifier,
{FullUrl}        nvarchar(260),
Type             tinyint,
MetaInfoTimeLastModified   datetime,
MetaInfo          varbinary(max),
Size              int,
TimeCreated       datetime,
TimeLastModified  datetime,
ETagVersion       int,
DocFlags          int,
{ListType}        int,
 tp_Name           nvarchar(38),
{ListTitle}       nvarchar(255),
CacheParseId      uniqueidentifier,
{GhostDirName}    nvarchar(256),
{GhostLeafName}   nvarchar(128),
 tp_Login          nvarchar(255),
CheckoutDate      datetime,
CheckoutExpires   datetime,
VirusStatus       int,
```
VirusInfo                      nvarchar(255),
SetupPathVersion               tinyint,
SetupPath                      nvarchar(255),
SetupPathUser                  nvarchar(255),
NextToLastTimeModified         datetime,
UIVersion                      int,
CheckinComment                 nvarchar(1023),
WelcomePageUrl                 nvarchar(260),
WelcomePageParameters          nvarchar(max),
tp Flags                       bigint,
Acl                            varbinary(max),
AnonymousPermMask              bigint,
DraftOwnerId                   int,
Level                          tinyint,
ParentVersion                  int,
TransformerId                  uniqueidentifier,
ParentLeafName                 nvarchar(128),
ProgId                         nvarchar(255),
DoclibRowId                    int,
{StreamSchema}                 tinyint,
InternalVersion                int

**Id:** The Document Identifier (section 2.2.1.1.2) of this contained document.

**{FullUrl}**: The URL in store-relative form for this document.

**Type:** The Document Store Type of the document.

**MetaInfoTimeLastModified:** A datetime with a timestamp in UTC format specifying the last time the MetaInfo column value of the document was changed. This value can be NULL if the MetaInfo column value of the document has never been changed.

**MetaInfo:** A metadict for the document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value can be NULL, and MUST be NULL if the MetaInfoTimeLastModified value is not more recent than the @ClientTimeStamp parameter.

**Size:** The number of bytes in the document stream of the document. This value MUST be NULL if the document does not have a document stream.

**TimeCreated:** A datetime with a timestamp in UTC format specifying when the document was created.

**TimeLastModified:** A datetime with a timestamp in UTC format specifying when the document was last modified.

**ETagVersion:** A counter used for internal conflict detection that is incremented any time a change is made to the document.

**DocFlags:** A Doc Flags value (section 2.2.2.3) describing the document. This value can be NULL if the document does not have any Doc Flags associated with it.

**{ListType}**: A packed combination of the List Base Type (section 2.2.1.2.11) and List Server Template (section 2.2.1.2.12) values of the list (1) containing this document, consisting of the List Server Template value multiplied by 256 and added to the value of the List Base Type.
**tp_Name:** The List Identifier (section 2.2.1.5) of the list (1) containing this document.

**{ListTitle}:** If the containing document is the root folder of the list (1) which contains this document, this contains the display name of the list (1). Otherwise, this value MUST be NULL.

**CacheParseId:** This value MUST be NULL.

**{GhostDirName}:** This value MUST be NULL.

**{GhostLeafName}:** This value MUST be NULL.

**tp_Login:** If this document is currently checked out, this is the login name of the user to whom it is checked out. In all other cases, this MUST be NULL.

**CheckoutDate:** A datetime with a timestamp in UTC format specifying when this document was checked out. This MUST be NULL if the document has never been checked out.

**CheckoutExpires:** A datetime with a timestamp in UTC format specifying when the short-term lock for this document will expire. If this date is in the past, this document SHOULD be treated as unlocked. This value can be NULL if no short-term lock has been placed on the document.

**VirusStatus:** A Virus Status value (section 2.2.1.2.17) specifying the current virus state of the document. This value can be NULL if the document has not been processed by a virus scanner.

**VirusInfo:** A string containing a provider-specific message returned by the virus scanner when it last processed the document. This value can be NULL if the document has not been processed by a virus scanner.

**SetupPathVersion:** If this is an uncustomized document, this specifies the setup path location that the SetupPath fragment is relative to. This value MUST NOT be NULL. The following are valid values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).</td>
</tr>
<tr>
<td>3</td>
<td>The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).</td>
</tr>
<tr>
<td>4</td>
<td>The SetupPath is relative to the install location of SharePoint Foundation 2010 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).</td>
</tr>
<tr>
<td>15</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Foundation 2013 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\15).</td>
</tr>
<tr>
<td>16</td>
<td>The SetupPath is relative to the install location of Microsoft SharePoint Server 2016 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\16).</td>
</tr>
</tbody>
</table>

**SetupPath:** If the document is now or once was uncustomized, this contains the setup path fragment relative to the base setup path described above by the SetupPathVersion value, where the document stream of the document can be found. This value can be NULL.

**SetupPathUser:** If the document is now or once was uncustomized, this contains the login name of the user who created the uncustomized document. This value can be NULL.

**NextToLastTimeModified:** The value of TimeLastModified from the previous time when the document was last saved. If the NextToLastTimeModified value matches the TimeLastModified
value when the change occurred and the client has a version of the document that it has successfully modified, the client can safely submit the document to the front-end Web server despite what appears to be an intervening edit to the document. This value MUST be NULL if the document has never been saved.

**UIVersion:** The **UI version** number for the document.

**CheckinComment:** An optional user-supplied description provided when the document is checked in or published. This value can be NULL.

**WelcomePageUrl:** If the document is a folder, this specifies an optional page to redirect to when the folder is requested with an **HTTP GET** operation. The URL is relative to the URL of the folder itself, and MUST be contained within that folder. Attempts to break out of the folder, such as "../../somepage.aspx", are not valid. This value can be NULL.

**WelcomePageParameters:** Contains optional URL parameters to specify as part of the **WelcomePageUrl** value. These parameters start at either the query string signifier '?' or the bookmark signifier '#'. This value can be NULL.

**tp_Flags:** The **List Flags** value (section 2.2.5) for the list (1) that contains the document.

**Acl:** The binary serialization of the **WSS ACL Format** (section 2.2.3.6) for the document. The **WSS ACL** is either explicitly defined for the document or inherited from the parent object of the document. This value can be NULL if a **WSS ACL** is not defined for the document.

**AnonymousPermMask:** A **WSS Rights Mask** (section 2.2.15) that indicates the permissions granted anonymous users or those users that have no specific permissions to the document. This value can be NULL if anonymous access to the document is not allowed.

**DraftOwnerId:** The **User Identifier** (section 2.2.1.13) of the user that published the document as a draft. This value MUST be NULL if the document is not a draft version.

**Level:** A **Publishing Level Type** value (section 2.2.2) specifying the publishing status of the document.

**ParentVersion:** If the document is a transformed version of another document, this is the UI version value from the original document. This value MUST be NULL if the document is not the product of a document transformation.

**TransformerId:** If the document is a transformed version of another document, this is the **GUID** of the agent that performed the transformation. This value MUST be NULL if the document is not the product of a document transformation.

**ParentLeafName:** If the document is a transformed version of another document, this is the **leaf name** of the original document. The original document MUST be in the same folder as the transformed document. If either is moved, the relationship is broken. This value MUST be NULL if the document is not the product of a document transformation.

**ProgId:** Specifies a preferred application to open the document. The **ProgId** value is used to distinguish between different applications that save files with a given file extension (for example, different editors for .HTML or .XML files). This value MUST be NULL if a **ProgId** was not specified when the document was saved.

**DoclibRowId:** The identifier for a row in a document library for the document. If the document is not contained in a list (1), this value MUST be NULL.

**tp_DefaultWorkflowId:** The **Workflow Identifier** (section 2.2.1.16) corresponding to the workflow to be invoked if the document is in a moderated list (1) and the document is submitted for approval as part of a check in.
**ListId:** The List Identifier (section 2.2.1.1.5) of the list (1) that contains the document. If the document is not contained in a list (1), this value MUST be NULL.

**ItemChildCount:** The number of list items for the list (1) that contains the document if the document is contained in a list (1).

**FolderChildCount:** The number of folders that exist in the list (1) that contains the document if the document is contained in a list (1).

**MetaInfoVersion:** A counter used for internal conflict detection that is incremented any time a change is made to the MetaInfo for this document.

**{CurVerMetaInfo}**: The value MUST be NULL.

**ContentVersion:** The version number of the document stream being returned.

**IsDirty:** A bit that specifies this document MUST have implementation-specific processing performed before its stream can be returned. If this document does not require processing, this value MUST be zero.

**{NextBSN}**: The current BLOB sequence number of the document being returned.

**{StreamSchema}**: The current stream schema of the document being returned.

**{InternalVersion}**: The internal version number of the document being returned.

### 3.1.5.48 proc_RenameUrl

The proc_RenameUrl stored procedure renames a document or folder within a specified site (2).

```sql
PROCEDURE proc_RenameUrl(  
    @SiteId                        uniqueidentifier,  
    @SubWebId                      uniqueidentifier,  
    @OldUrl                        nvarchar(260),  
    @NewUrl                        nvarchar(260),  
    @UserId                        int,  
    @AppPrincipalId                int,  
    @ThresholdRowCount             int,  
    @NewDoclibRowIdInput           int,  
    @MaxNewRowsInput               int,  
    @RenameFlags                   int                = 0,  
    @PutFlags                      int                = 0,  
    @ReturnFlags                   int                = 0,  
    @AttachmentOpOldUrl            int                = 3,  
    @AttachmentOpNewUrl            int                = 3,  
    @OldItemld                     int                = NULL OUTPUT,  
    @ParseDocNow                   tinyint            = NULL OUTPUT,  
    @FailedUrl                     nvarchar(260)      = NULL OUTPUT,  
    @RequestGuid                   uniqueidentifier   = NULL OUTPUT
);
```

**@SiteId:** The site collection identifier (section 2.2.1.1.9) of the site collection containing the document or folder.

**@SubWebId:** The site identifier of the site (2) containing the document or folder.

**@OldUrl:** The store-relative URL of the document or folder to be moved.

**@NewUrl:** The target store-relative URL of the document or folder after the move.

**@UserId:** The user identifier of the user who is requesting the move.
@AppPrincipalId: The app principal identifier of the app that is requesting the operation. If the operation was not requested by an app then it MUST be 0.

@ThresholdRowCount: If the @OldUrl parameter is a folder and this parameter is greater than 0 and the sum total of documents and folders contained within the @OldUrl parameter exceeds this parameter's value, then this stored procedure MUST NOT complete successfully.

@NewDoclibRowIdInput: The next available row number in the new list where this item is being copied.

@MaxNewRowsInput: The number of items that are being renamed as part of this rename operation.

@RenameFlags: A bit field determining document rename options. This can have one or more flags set. The only valid values of the @RenameFlags bits are specified in Rename Flags (section 2.2.2.8).

@PutFlags: A bit field determining document change options. This can have one or more flags set. The only valid values of the @PutFlags bits are specified in Put Flags Type (section 2.2.2.7).

@ReturnFlags: A bit field determining the type of information requested. This can have one or more flags set. The only valid bit values of @ReturnFlags are specified as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>Return information about the moved documents.</td>
</tr>
<tr>
<td>0x02</td>
<td>Return information about backward links referencing the moved documents.</td>
</tr>
</tbody>
</table>

@AttachmentOpOldUrl: An Attachments Flag (section 2.2.1.2.1) value for the document URL.

@AttachmentOpNewUrl: An Attachments Flag value for the destination URL.

@OldItemId: If this parameter is supplied, it will be set to the row number that was being used by the previous document (the URL specified by the @OldUrl parameter). If the @OldUrl parameter is a folder, then this parameter MUST be ignored.

@ParseDocsNow: An output parameter set to 1 to indicate that the moved document or documents in the folder MUST be parsed again. Reparsing is necessary when the document properties need to be scanned, for example, when the document is moved across lists, or when the document's file extension is modified. If this output parameter is supplied, then it will be set to 0 indicating that the document or documents in the folder MUST NOT be parsed again.

@FailedUrl: An output parameter indicating the URL at which the move operation failed. MUST be set to NULL if the move was successful.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_RenameUrl stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>2</td>
<td>File not found: Parent of target URL not found.</td>
</tr>
<tr>
<td>3</td>
<td>Path not found: Item not found at specified URL, site collection and site (2).</td>
</tr>
<tr>
<td>5</td>
<td>Access denied: The User specified in @UserId lacks the necessary privileges.</td>
</tr>
<tr>
<td>15 or</td>
<td>Cannot move to, or from, a forms folder in a list or document library.</td>
</tr>
<tr>
<td>51</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>33</td>
<td>Cannot move a folder which contains checked out documents.</td>
</tr>
<tr>
<td>34</td>
<td>Cannot rename a folder within a List or move a folder out of a list when the list is not a document library.</td>
</tr>
<tr>
<td>36</td>
<td>The rename operation failed to complete successfully because the value specified for the <code>@ThresholdRowCount</code> parameter exceeds the sum total of documents and folders contained within the URL specified by the <code>@OldUrl</code> parameter.</td>
</tr>
<tr>
<td>50</td>
<td>The document is a site (2), but <code>@RenameFlags</code> does not indicate that a Site move is requested.</td>
</tr>
<tr>
<td>80</td>
<td>Target URL is a document with a Document Store Type (section 2.2.2.4) of 0 (File), but <code>@PutFlags</code> value does not indicate a request to overwrite it.</td>
</tr>
<tr>
<td>87</td>
<td>Bad parameter: Unspecified Error prevented the requested move.</td>
</tr>
<tr>
<td>130</td>
<td>Cannot move to, or from, an image or a thumbnail file in an Image Library.</td>
</tr>
<tr>
<td>138</td>
<td>Cannot move folder across lists.</td>
</tr>
<tr>
<td>144</td>
<td>Not an overwrite request and URL is a directory or site (2), or other invalid combination.</td>
</tr>
<tr>
<td>161</td>
<td>Cannot move folder across sites.</td>
</tr>
<tr>
<td>190</td>
<td>Cannot move folder containing thicket.</td>
</tr>
<tr>
<td>206</td>
<td>Target URL is too long.</td>
</tr>
<tr>
<td>214</td>
<td>Cannot move thicket.</td>
</tr>
<tr>
<td>1150</td>
<td>Concurrency violation.</td>
</tr>
<tr>
<td>1358</td>
<td>Internal database corruption.</td>
</tr>
<tr>
<td>1359</td>
<td>Unknown internal error.</td>
</tr>
<tr>
<td>8398</td>
<td>There was an error moving a list. At least one list could not be deleted.</td>
</tr>
</tbody>
</table>

This stored procedure MUST return either zero, one or two result sets, depending on the values for input parameters as described for each result set.

### 3.1.5.48.1 Rename Result Set

The Rename Result Set returns basic information about the old and new URLs for all moved (renamed) Documents. When renaming a container object, affected items in the container are included in the Rename Result Set. The Rename Result Set MUST be returned only when requested (when bit 0x01 is set in `@ReturnFlags`). If the Rename Result Set is returned, it MUST return one row for each document which was renamed during the operation.

```
{OldUrlDirName}    nvarchar(256),
{OldUrlLeafName}   nvarchar(128),
{NewUrlDirName}    nvarchar(256),
{NewUrlLeafName}   nvarchar(128),
{Type}            int;
```

- `{OldUrlDirName}`: The directory name of the document before rename.
- `{OldUrlLeafName}`: The leaf name of the document before rename.
- `{NewUrlDirName}`: The directory name of the document after rename.
{NewUrlLeafName}: The leaf name of the document after rename.

{Type}: The Document Store Type (section 2.2.4) of the renamed document.

3.1.5.48.2 Backward Link Result Set

The Backward Link Result Set returns the URL of each document containing a backward link to the moved (renamed) document(s). The Backward Link Result Set MUST be returned only when requested (bit 0x2 is set in @ReturnFlags and bit 0x4 is set in @RenameFlags.) If the Backward Link Result Set is returned, it MUST return one row for each item containing a backward link to any of the renamed document(s).

```
DocUrl    nvarchar(260),
```

DocUrl: The store-relative form URL to the document holding the backward link to the renamed document.

3.1.5.49 proc_SecAddPrincipalToRole

The proc_SecAddPrincipalToRole stored procedure is invoked to add a security principal to a role defined within a site collection.

```
PROCEDURE proc_SecAddPrincipalToRole(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @ScopeId                       uniqueidentifier,
    @RoleId                        int,
    @UserId                        int,
    @AddChangeLog                  bit = 0,
    @ReturnAuditMask               bit = 1,
    @AddToCurrentScopeOnly         bit = 0,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);```

@SiteId: The site collection identifier of the site collection containing the role and security principal (2).

@WebId: The site identifier of the site (2) containing the role and security principal.

@ScopeId: The scope identifier of the security scope containing the role.

@RoleId: Specifies the role identifier of the role definition that the security principal is added into. @RoleId MUST correspond to a valid role.

@UserId: The user identifier for the security principal (2) to be added to the specified role. This value MUST refer to an existing user identifier for the specified site collection.

@AddChangeLog: A bit flag specifying whether to update the change log. A value of 1 indicates that the change log MUST be updated. The default value of 0 indicates that the change log MUST NOT updated.

@ReturnAuditMask: A bit flag specifying whether to return Audit Flags (section 2.2.2.1) data for the specified site (2). The default value of 1 indicates that Audit Flags data MUST be returned. A value of 0 indicates that Audit Flags data MUST NOT be returned.

@AddToCurrentScopeOnly: A bit flag specifying whether the principal will be added only to the given security scope. The default value of 0 indicates that security principal MUST be added to all security
scopes up to the security scope of web indicated by @WebId. A value of 1 indicates that security principal MUST be added only to the security scope indicated by @ScopeId.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecAddPrincipalToRole stored procedure returns an integer return code which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The site (2) specified by @WebId is not valid for membership permissions or there are no roles defined for this site.</td>
</tr>
</tbody>
</table>

The proc_SecAddPrincipalToRole stored procedure MUST return one result set if the @ReturnAuditMask value is set to 1, and MUST NOT return a result set if the value is set to 0.

### 3.1.5.49.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) set for the site (2). The Site Audit Mask Result Set MUST be returned if the @ReturnAuditMask value is set to 1. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).

### 3.1.5.50 proc_SecAddRoleDef

The proc_SecAddRoleDef stored procedure creates a new role definition for a specified site (2) within a site collection.

```
PROCEDURE proc_SecAddRoleDef(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @Title                         nvarchar(255),
    @Description                   nvarchar(512),
    @Hidden                        bit,
    @RoleOrder                     int,
    @Type                          tinyint,
    @PermMask                      tPermMask,
    @IdToCreate                    int,
    @RoleDefId                     int = NULL OUTPUT,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT);
```

@SiteId: The site collection identifier of the site collection to add the role definition to.

@WebId: The site identifier of the site (2) to add the role definition to. The site (2) MUST have its own scope specifying unique permissions.

@Title: The title of the role definition for display in the front-end Web server. This MUST be a value which does not match an existing role definition title within the site (2), and MUST NOT be NULL or an empty string.

@Description: The description of the role definition for display in the front-end Web server.

@Hidden: A bit flag specifying whether or not the role definition is to be displayed by the front-end Web server. This value MUST NOT be NULL. If this parameter is set to 1, then the front-end Web server MUST NOT display the role definition.
@RoleOrder: An integer value specifying the relative position in which this role definition is to be displayed in the front-end Web server. Multiple role definitions can have the same @RoleOrder value. This value MUST NOT be NULL.

@Type: The role definition type value for the role definition to be added.

@PermMask: A WSS Rights Mask (section 2.2.2.15) specifying the rights to grant to the role definition. The WSS Rights Mask associated with the role definition MUST be set to this value.

@IdToCreate: The role identifier to assign to the new role definition. If this parameter is NULL, a new value MUST be assigned and returned in @RoleDefId. This parameter MUST NOT be a role identifier which already exists within the site collection.

@RoleDefId: The role identifier of the created role definition, returned as an output parameter.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecAddRoleDef stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>80</td>
<td>The @IdToCreate or @Title already exists or is NULL for a role definition within the site collection.</td>
</tr>
<tr>
<td>1816</td>
<td>The limit for the maximum number of role definitions in the specified site (2) has been reached.</td>
</tr>
</tbody>
</table>

Upon successful execution, the proc_SecAddRoleDef stored procedure MUST return a single result set.

3.1.5.50.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the site (2). The Site Audit Mask Result Set MUST be returned on successful completion and MUST contain one row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).

3.1.5.51 proc_SecAddUser

The proc_SecAddUser stored procedure is invoked to add a new entry for a security principal to the UserInfo Table (section 2.2.6.10) that contains descriptive properties and security information about security principals (2), and is stored in the back-end database server.

PROCEDURE proc_SecAddUser(
    @SiteId uniqueidentifier,
    @SystemId varbinary(512),
    @ExternalToken varbinary(max),
    @ExternalTokenTime datetime,
    @IsDomainGroup bit,
    @IsActive bit,
    @AccountType int,
    @Login nvarchar(255),
    @Title nvarchar(255),
    @Email nvarchar(255),
    @Notes nvarchar(1023),
    @MobilePhone nvarchar(127),
    @Flags int,
    @MembershipWebId uniqueidentifier = NULL,
    @IncrementUserCount bit = 0,
    @ImportDeleted bit = 0,
@AddedToTable bit OUTPUT,
@UserIdOut int OUTPUT,
@LoginOut nvarchar(255) OUTPUT,
@TitleOut nvarchar(255) OUTPUT,
@EmailOut nvarchar(255) OUTPUT,
@NotesOut nvarchar(1023) OUTPUT,
@MobilePhoneOut nvarchar(127) OUTPUT,
@DeletedOut bit OUTPUT,
@NeedToAddToList bit OUTPUT,
@RequestGuid uniqueidentifier = NULL OUTPUT
);

@SiteId: The site collection identifier of the site collection to associate with the principal (1).

@SystemId: The SystemID of the principal to be added or updated. If a user exists with the specified SystemID, its record will be updated. Otherwise, a new user will be added with the specified SystemID.

@ExternalToken: An External Group Token (section 2.2.3.2) specifying information on the principal's domain group membership, derived from an external security provider.

@ExternalTokenTime: A datetime in UTC format specifying the most recent time the @ExternalToken value was updated.

@IsDomainGroup: A bit flag specifying whether the principal to be added is a domain group. If this is set to 1, the principal being added is a domain group. If this is set to 0, then the principal is a user. This parameter MUST NOT be NULL.

@IsActive: A bit flag specifying if the principal is a user not marked as deleted in the site collection. When set to 1, the principal is a user not marked as deleted in the site collection. If this is set to 0, then the principal is a user marked as deleted in the site collection. This parameter MUST NOT be NULL.

@AccountType: The account type of user. If @SystemId isn't null or NULLSID (value is S-1-0-0, which represents a group (2) with no members), and @AccountType is 2, then the added user's User Identifier represents an app account identifier (1073741822). All other values will be ignored.

@Login: The login name of the principal to be added. This parameter MUST NOT be NULL.

@Title: The display name of the principal to be added. This parameter MUST NOT be NULL.

@Email: The email address of the principal to be added. This parameter MUST NOT be NULL.

@Notes: A string containing notes about the principal to be added. This parameter MUST NOT be NULL.

@MobilePhone: The mobile phone of the principal to be added.

@Flags: AUserInfo Flags (section 2.2.2.12) specifying the principal's options.

@MembershipWebId: The site identifier of the site (2) within the same site collection that the principal will be added as a member of.

@IncrementUserCount: A bit flag specifying whether to increment the user count of the site collection. When this parameter is set to 1, the user count in the site collection MUST be incremented.

@ImportDeleted: A bit flag specifying whether the principal is to be added as deleted. When this parameter is set to 1, the user information for the principal MUST be marked as deleted. The deleted state is set within the UserInfo Table, rather than dropping entries from the UserInfo Table, to preserve list item ownership information.
@AddedToTable: An output parameter indicating that an entry for the principal has been added to the UserInfo Table. Its value MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The principal already exists in the UserInfo Table.</td>
</tr>
<tr>
<td>1</td>
<td>The user information for the principal has been added to the UserInfo Table.</td>
</tr>
</tbody>
</table>

@UserIdOut: An output parameter which MUST contain the user identifier of the added principal on successful completion.

@LoginOut: An output parameter which MUST contain the login name of the added principal on successful completion.

@TitleOut: An output parameter which MUST contain the display name of the added principal on successful completion. It MUST be NULL otherwise.

@EmailOut: An output parameter which MUST contain the email address of the added principal on successful completion. It MUST be NULL otherwise.

@NotesOut: An output parameter which MUST contain the notes about the added principal on successful completion. It MUST be NULL otherwise.

@MobilePhoneOut: An output parameter which MUST contain the mobile phone of the added principal on successful completion. It MUST be NULL otherwise.

@DeletedOut: An output parameter which can indicate the deleted state of the principal. Its value MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The user information has not been marked as deleted.</td>
</tr>
<tr>
<td>1</td>
<td>The user information has been marked as deleted.</td>
</tr>
</tbody>
</table>

@NeedtoAddtoList: An output parameter which can indicate if this user is already in the userinfo list.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The user is already in the UserInfo list, and does not need to be added again.</td>
</tr>
<tr>
<td>1</td>
<td>The user is not in the UserInfo list, it needs to be added to UserInfo list later.</td>
</tr>
</tbody>
</table>

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_SecAddUser` stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>80</td>
<td>A principal with the same login name but a different SystemID already exists.</td>
</tr>
<tr>
<td>212</td>
<td>The site collection is locked against write operations occurring.</td>
</tr>
<tr>
<td>1359</td>
<td>An internal error has occurred.</td>
</tr>
<tr>
<td>1816</td>
<td>The maximum number of users that can be added to a site (2) has been exceeded and no additional...</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>users can be added to the site.</td>
</tr>
</tbody>
</table>

The **proc_SecAddUser** stored procedure MUST NOT return any result sets.

### 3.1.5.52 proc_SecAddUserToSiteGroup

The **proc_SecAddUserToSiteGroup** stored procedure is invoked to add a user to a permission level in the site collection. The user is added to the permission level and the root site of the site collection, the user's User Token and status as active user is updated, and the change is logged to the change log and site audit log.

```sql
PROCEDURE proc_SecAddUserToSiteGroup(
    @SiteId                        uniqueidentifier,
    @GroupId                       int,
    @UserIDToBeAdded               int,
    @UserID                        int,
    @AppPrincipalId                int,
    @SiteAdmin                     bit,
    @BelongsToGroup                bit,
    @GroupOwnerId                  int,
    @CurrentUserIsOwner            bit,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

**@SiteId**: The site collection identifier (section 2.2.1.1.9) for the site collection that contains the permission level specified by **@GroupId**.

**@GroupId**: A permission level identifier for the permission level that the user specified by **@UserIDToBeAdded** refers to. This value MUST refer to an existing permission level within the site collection specified by **@SiteId**.

**@UserIDToBeAdded**: A user identifier for the user being added to the permission level specified by **@GroupId**. This value MUST refer to an existing user identifier for the site collection specified by **@SiteId**.

**@UserID**: The user identifier for the user who is adding the user specified by **@UserIDToBeAdded** to the permission level specified by **@GroupId**. This value MUST refer to an existing user Identifier in the site collection specified by **@SiteId**.

**@AppPrincipalId**: The app principal identifier of the app that is requesting the operation. If the operation was not requested by an app then it MUST be set to 0.

**@SiteAdmin**: If this parameter is set to 1, the user specified by **@UserID** is a site collection administrator. If this parameter is set to 0, then the user specified by **@UserID** is not a site collection administrator.

**@BelongsToGroup**: If this parameter is set to 1, then the user specified by **@UserID** is a member of the permission level. If this parameter is set to 0, then the user specified by **@UserID** is not a member of the permission level.

**@GroupOwnerId**: The user identifier for the owner of the permission level specified by **@GroupId**. **@GroupOwnerId** MUST refer to an existing owner, which can be either a user or another permission level.

**@CurrentUserIsOwner**: If this parameter is set to 1, then the user specified by **@GroupOwnerId** is the owner of the permission level specified by **@GroupId**. If this parameter is set to 0, the user is not the owner.
@RequestGuid: The optional request identifier for the current request.

Return Values: The **proc_SecAddUserToSiteGroup** stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution or the user already belongs to the permission level.</td>
</tr>
<tr>
<td>5</td>
<td>The user specified by @UserID does not have rights to modify the membership of the permission level specified by @GroupID.</td>
</tr>
</tbody>
</table>

The **proc_SecAddUserToSiteGroup** stored procedure MUST NOT return any result sets.

### 3.1.5.53 **proc_SecAddWebMembership**

The **proc_SecAddWebMembership** stored procedure is invoked to associate an existing user within a site collection to a site (2) within the same site collection.

```sql
PROCEDURE proc_SecAddWebMembership(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @UserId                        int,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection containing the requested site (2) specified by @WebId and requested user identifier specified by @UserId.

@WebId: The site identifier of the site (2) to add a new user to. This value MUST refer to an existing site identifier within the site collection specified by @SiteId.

@UserId: The user identifier for the user to be added to membership of the site (2). This value MUST refer to an existing user identifier within the site collection specified by @SiteId.

@RequestGuid: The optional request identifier for the current request.

Return Values: The **proc_SecAddWebMembership** stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The site (2) specified by @WebId was not found, or the ancestor site with unique permissions that defines the scope for the site specified by @WebId was not found.</td>
</tr>
</tbody>
</table>

The **proc_SecAddWebMembership** stored procedure MUST NOT return any result sets.

### 3.1.5.54 **proc_SecChangeToInheritedList**

The **proc_SecChangeToInheritedList** stored procedure is invoked to change the scope of the specified list to the specified site (2) and remove any role assignments and permission settings specific to the list.

The permissions of any list items contained in the specified list which have unique permissions MUST NOT be changed.
PROCEDURE proc_SecChangeToInheritedList(
    @SiteId uniqueidentifier,
    @WebId uniqueidentifier,
    @ListId uniqueidentifier,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);}

@SiteId: The site collection identifier of the site collection.

@WebId: The site identifier of the site (2) containing the list specified by @ListId.

@ListId: The list identifier of the list.

@RequestGuid: The optional request identifier for the current request.

Return Values: This procedure returns an integer return code which MUST be 0, and MUST return one result set on successful completion. If this procedure encounters a failure, then no result set is returned.

3.1.5.54.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the specified site (2). The Site Audit Mask Result Set MUST be returned on successful completion, and MUST return a single row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).

3.1.5.55 proc_SecChangeToInheritedWeb

The proc_SecChangeToInheritedWeb stored procedure changes a site (2) from having its own unique permissions to instead use the permissions inherited from its nearest ancestor with unique permissions. When a site (2) is changed to have inherited permissions, all role definitions and permissions for each list and document library are set to inherit from the security scope that the site uses.

PROCEDURE proc_SecChangeToInheritedWeb(
    @SiteId uniqueidentifier,
    @WebId uniqueidentifier,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);}

@SiteId: The site collection identifier for the site collection containing the site (2) to be changed specified by @WebId.

@WebId: The site identifier for the site (2) to change to use inherited permissions.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecChangeToInheritedWeb stored procedure returns an integer return code that MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The site (2) was not found, the site does not have unique permissions, or an ancestor site with unique permissions was not found.</td>
</tr>
</tbody>
</table>
The **proc_SecChangeToInheritedWeb** stored procedure MUST return two result sets on successful execution. If the stored procedure encounters a failure, then no result sets are returned.

### 3.1.5.55.1 Inherited Site Result Set

The **Inherited Site Result Set** returns the scope identifier and site identifier that the site (2) specified by @WebId now inherits its permissions from. The **Inherited Site Result Set** MUST be returned on successful completion and MUST contain one row.

<table>
<thead>
<tr>
<th>ScopeId</th>
<th>uniqueidentifier,</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ParentWebPermAncestor}</td>
<td>uniqueidentifier;</td>
</tr>
</tbody>
</table>

**ScopeId**: The scope identifier for the scope that the changed site (2) specified by @WebId now inherits from.

**{ParentWebPermAncestor}**: The site identifier for an ancestor site that the changed site (2) specified by @WebId now derives permissions from.

### 3.1.5.55.2 Site Audit Mask Result Set

The **Site Audit Mask Result Set** contains information about the **Audit Flags** (section 2.2.2.1) associated with the site (2) specified by @WebId. The **Site Audit Mask Result Set** MUST be returned on successful completion and MUST contain a single row.

The **Site Audit Mask Result Set** is defined in the Common Result Sets **Site Audit Mask Result Set** (section 2.2.4.22).

### 3.1.5.56 proc_SecChangeToUniqueScope

The **proc_SecChangeToUniqueScope** stored procedure sets a securable object such as a site (2), list (1), or document library to use its own unique security scope, instead of inheriting its security scope from the first ancestor with uniquely permissions.

```sql
PROCEDURE proc_SecChangeToUniqueScope(  
  @SiteId uniqueidentifier,  
  @WebId uniqueidentifier,  
  @OldScopeId uniqueidentifier,  
  @CopyFromScopeId uniqueidentifier,  
  @Url nvarchar(260),  
  @DocId uniqueidentifier,  
  @bIsWeb bit,  
  @UserId int,  
  @CopyAnonymousMask bit,  
  @CopyRoleAssignments bit,  
  @ClearSubScopes bit,  
  @bBreakBySiteOwner bit,  
  @ReturnAuditMask bit,  
  @MaxScopeInList int,  
  @NewScopeId uniqueidentifier = NULL OUTPUT,  
  @RequestGuid uniqueidentifier = NULL OUTPUT  
);  
```

**@SiteId**: The **site collection identifier** of the **site collection** containing the securable object specified by @Url or @DocId to be set to use a unique security scope.

**@WebId**: The **site identifier** of the site (2) that is or contains the securable object.

**@OldScopeId**: The **scope identifier** for the original security scope of the securable object specified by @Url or @DocId.
@CopyFromScopeId: The scope identifier for a security scope to copy the administrator role, **anonymous user** permissions, and **role assignments** from for use as the new security scope. This parameter MUST NOT be NULL.

@Url: The store-relative form URL for the securable object. The securable object MUST be specified by the @Url or the @DocId parameter. The @Url parameter MUST be NULL to specify the securable object with the @DocId parameter.

@DocId: The document identifier of the securable object. The @DocId parameter MUST be ignored and can be NULL if @Url specifies the securable object.

@bIsWeb: A bit flag specifying whether the securable object is a site (2). If this parameter is set to one, the securable object is a site (2), and all subsites which inherit their security scope from the site (2) specified by @SiteId MUST have their inheritances changed to the new security scope. This parameter MUST be set to zero if @Url does not point to a site (2), and this parameter MUST be set to one if @Url points to a site (2).

@UserId: Specifies the principal to be added to the new security scope in the administrator role, unless overridden by the @bBreakBySiteOwner parameter or the @CopyFromScopeId parameter.

@CopyAnonymousMask: A bit flag specifying whether to copy anonymous user permissions from the @CopyFromScopeId parameter into the new security scope. If this parameter is set to one, the permissions for anonymous user access MUST be copied from the @CopyFromScopeId parameter into the new security scope.

@CopyRoleAssignments: A bit flag specifying whether to copy the role assignments from the @CopyFromScopeId parameter into the new security scope. If this parameter is set to one, the role assignments MUST be copied from the @CopyFromScopeId parameter into the new security scope. If both @bBreakBySiteOwner and @CopyRoleAssignments are set to one, the setting of @CopyRoleAssignments will take precedence. The role assignments are copied from the security scope provided in @CopyFromScopeId.

@ClearSubScopes: A bit flag specifying whether to change every site (2), list (1) and document library under the URL to the new security scope or to change only inheriting ones to the new security scope. If this parameter is set to one, every site (2), list (1) and document library under the URL MUST be changed to the new security scope, including no inheriting ones.

@bBreakBySiteOwner: A bit flag specifying whether to use the site (2) owner in the administrator role instead of the principal specified by @UserId. If this parameter is set to one, the site (2) owner MUST be added to the administrator role in the new security scope definition. If both @bBreakBySiteOwner and @CopyRoleAssignments are set, the setting of @CopyRoleAssignments will take precedence. The role assignments are copied from the security scope provided in @CopyFromScopeId.

@ReturnAuditMask: A bit flag specifying whether to return a Site Audit Mask Result Set (section 3.1.5.56.1). If this parameter is set to one, a Site Audit Mask Result Set MUST be returned on successful completion.

@MaxScopeInList: An integer value specifying the maximum number of unique security scopes under a list (1). If a list item's security scope is changed and the list (1) including this list item already has @MaxScopeInList unique security scopes, the stored procedure MUST return an error using the integer return code 1340.

@NewScopeId: The Scope Identifier (section 2.2.1.1.8) generated for the new security scope, returned as an output parameter.

@RequestGuid: The optional request identifier for the current request.

**Return Values:** The proc_SecChangeToUniqueScope stored procedure returns an integer return code which MUST be listed in the following table.
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The specified securable object was not found at the specified location, the @OldScopeId does not match the scope identifier of the securable object, or the securable object has unique permissions.</td>
</tr>
<tr>
<td>1340</td>
<td>The security scope of a list item is changed and the list (1) including this list item already has @MaxScopeInList unique security scopes.</td>
</tr>
</tbody>
</table>

If @ReturnAuditMask has a value of one, the `proc_SecChangeToUniqueScope` stored procedure MUST return a single result set on successful completion; otherwise, zero result sets MUST be returned.

### 3.1.5.56.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the site (2). If @ReturnAuditMask has a value of 1, the Site Audit Mask Result Set MUST be returned on successful completion, and MUST contain only one row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).

### 3.1.5.57 proc_SecCheckDeletedAccounts

The `proc_SecCheckDeletedAccounts` stored procedure is invoked to check if a login name exists in the site collection.

```sql
PROCEDURE proc_SecCheckDeletedAccounts(
    @SiteId       uniqueidentifier,
    @Login        nvarchar(255),
    @RequestGuid  uniqueidentifier = NULL OUTPUT
);
```

- **@SiteId**: The site collection identifier for the site collection.
- **@Login**: The login name of the principal as specified by the security provider in use. This parameter MUST NOT be NULL.
- **@RequestGuid**: The optional request identifier for the current request.

Return Values: The `proc_SecCheckDeletedAccounts` stored procedure returns an integer return code, which MUST be 0, and it MUST return a single result set.

### 3.1.5.57.1 Login Result Set

If a valid login name is found in the site collection, the Login Result Set MUST return one row containing the login name for each time @Login is found for the site collection specified by @SiteId within the UserInfo Table (section 2.2.6.10); otherwise zero rows are returned.

```sql
tp_Login        nvarchar(255);
```

- **tp_Login**: The login name for the user.

### 3.1.5.58 proc_SecCloneRoleDefinitions

The `proc_SecCloneRoleDefinitions` stored procedure creates a copy of the current role definition for a site (2). After successful execution, the site (2) will have its own copy of the role definition and
unique role assignments, and the site (2) and all subsites that inherit permissions will use the new role definition.

```sql
PROCEDURE proc_SecCloneRoleDefinitions(
    @SiteId uniqueidentifier,
    @WebId uniqueidentifier,
    @CopyRoleAssignments bit,
    @UserId int,
    @NewScopeId uniqueidentifier OUTPUT,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection containing the site (2).

@WebId: The site identifier of the site (2) which will have its current role definition copied.

@CopyRoleAssignments: Specifies whether to keep the current role assignments. If this parameter is set to 1, the current role assignments will kept. If this parameter is set to 0, then the user specified by @UserId will be added to the Administrator Role, and everyone else will be removed from all roles. @CopyRoleAssignments MUST NOT be NULL.

@UserId: The user identifier of the current user. @UserId is assigned to the administrator role when @CopyRoleAssignments is set to 0. This value MUST refer to an existing user identifier for the specified site collection.

@NewScopeId: An output parameter which contains a scope identifier for the security scope of the site (2). If the site (2) already has unique role assignments before this call, then a new security scope MUST NOT be generated, and the output parameter MUST be the original scope identifier of the site. If the site (2) does not have unique role assignments before this call, then a new security scope MUST be generated for this site, and the scope identifier of the new security scope MUST be returned in the output parameter.

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_SecCloneRoleDefinitions` stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The site (2) was not found. Either the site (2) specified by @SiteId/@WebId does not exist, or the site already has its own role definition.</td>
</tr>
<tr>
<td>80</td>
<td>The site (2) has a role definition with the same title as one of the role definitions that will be cloned.</td>
</tr>
</tbody>
</table>

The `proc_SecCloneRoleDefinitions` stored procedure MUST return either one or two result sets on successful execution, both of which used the same result set definition.

### 3.1.5.58.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains the information about the Audit Flags (section 2.2.2.1) associated with the specified site (2). On successful execution, the Site Audit Mask Result Set MUST be returned only once if @CopyRoleAssignments is set to 1. If @CopyRoleAssignments is set to 0, then the Site Audit Mask Result Set MUST be returned twice on successful execution.

The Site Audit Mask Result Set MUST contain a single row. The Site Audit Mask Result Set is defined in Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).
3.1.5.59 proc_SecCreateSiteGroup

The proc_SecCreateSiteGroup stored procedure is invoked to add a new permission level to a site collection.

```sql
PROCEDURE proc_SecCreateSiteGroup(
    @SiteId uniqueidentifier,
    @Title nvarchar(255),
    @Description nvarchar(512),
    @OwnerID int,
    @OwnerIsUser bit,
    @AppPrincipalId int,
    @FirstMemberId int,
    @UseExisting bit,
    @SelfOwner bit,
    @Flags int,
    @GroupId int OUTPUT,
    @RequestGuid uniqueidentifier = NULL OUTPUT
)
```

@SiteId: The site collection identifier (section 2.2.1.9) of the site collection containing the new permission level to be created.

@Title: The title of the new permission level. This parameter MUST NOT be NULL.

@Description: A description of the new permission level.

@OwnerID: A user identifier or site group identifier for the new site group's owner.

@OwnerIsUser: A bit flag specifying whether the site group owner specified by @OwnerId is a user or a site group.

- When @OwnerIsUser is set to 1, the new site group owner is a user in the site collection.
- When @OwnerIsUser is set to 0, the owner is a site group.

@AppPrincipalId: The app principal identifier of the app that is requesting the operation. If the operation was not requested by an app then it MUST be set to 0.

@FirstMemberId: A user identifier for the first member of the new site group. This value MUST correspond to an existing user identifier or be NULL. If this value is NULL then the site group MUST be created with no members.

@UseExisting: A bit flag specifying whether to return the site group identifier of an existing site group with a title matching @Title or to create a new site group.

- When @UseExisting is set to 1, the procedure MUST return an existing site group with the same title as the @Title input parameter, and if such a site group is not found, then it MUST create a new site group.
- When @UseExisting is set to 0, the procedure MUST return a failure code value of 80 if an existing site group with a matching title is found, and if such a site group is not found, then it MUST create a new site group.

@SelfOwner: A bit flag specifying whether or not the site group is its own owner. When @SelfOwner is set to 1, the site group MUST be set as its own owner, and the values of @OwnerId and @OwnerIsUser MUST be ignored.

@Flags: Contains a set of flags specifying properties about the site group. This parameter MUST NOT be NULL and MUST be one of the values listed in the following table.
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Allow anyone to view the membership of the site group.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Only allow members of the site group to view the membership.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Allow members of the site group to edit the membership of the site group.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>Allow users to request membership in this site group, and allow users to request to leave the site group. Requests MUST be sent to the e-mail address specified by RequestEmail.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>Automatically accept user requests to join or leave the site group. This bit MUST be set only when the bit 0x00000004 is also set.</td>
</tr>
<tr>
<td>0x00000010</td>
<td>The site group is hidden in the user interface.</td>
</tr>
</tbody>
</table>

@GroupID: An output parameter holding the integer site group identifier for the newly created or existing site group. If the procedure fails, then the value of this parameter SHOULD be ignored.

@RequestGuid: The optional request identifier for the current request.

**Return Values:** The `proc_SecCreateSiteGroup` stored procedure returns an integer return code, which MUST be included in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>80</td>
<td>Failed to create the site group because a site group with an identical title exists in the site collection.</td>
</tr>
</tbody>
</table>

The `proc_SecCreateSiteGroup` stored procedure MUST NOT return any result sets.

### 3.1.5.60 proc_SecDecCurrentUsersCount

The `proc_SecDecCurrentUsersCount` stored procedure is invoked to reduce by one the total number of users in the specified site collection when configured to use Active Directory account creation mode. `proc_SecDecCurrentUsersCount` does not delete or update any user information.

```
PROCEDURE proc_SecDecCurrentUsersCount(
    @SiteId                        uniqueidentifier,
    @RequestGuid                   uniqueidentifier = NULL    OUTPUT
);
```

@SiteID: The Site Collection Identifier of the site collection.

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_SecDecCurrentUsersCount` stored procedure returns an integer return code, which MUST be 0.

The `proc_SecDecCurrentUsersCount` stored procedure MUST return no result sets.

### 3.1.5.61 proc_SecGetAccountStatus

The `proc_SecGetAccountStatus` stored procedure provides status information for a site collection’s users, which are not marked as deleted, and match the specified login name or email address.

```
PROCEDURE proc_SecGetAccountStatus(
    @SiteId                        uniqueidentifier,
    );
```

355 / 484
@SiteId: The Site Collection Identifier of the site collection containing the requested Users.

@Login: The login name of a User to be matched.

@Email: The email address of a User to be matched. If this parameter is an empty string, it is ignored.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetAccountStatus stored procedure MUST return an integer return code of 0.

The proc_SecGetAccountStatus stored procedure MUST return one result set as follows.

3.1.5.61.1 Account Status Result Set

The Account Status Result Set returns account information for the non-deleted Users matching the specified login name or email address. This result set MUST be returned and MUST contain one row for each User which is not a domain group. The Account Status Result Set MUST contain no rows if no matching Users are found.

    tp_Login                        nvarchar(255),
    tp_Email                        nvarchar(255),
    tp_SystemID                     varbinary(512);

    tp_Login: The login name of the matching User.
    tp_Email: The email address of the matching User.
    tp_SystemID: The SystemId of the matching User.

3.1.5.62 proc_SecGetAclFromScope

The proc_SecGetAclFromScope stored procedure is invoked to obtain the Access Control List and the anonymous User permissions for a given scope.

    PROCEDURE proc_SecGetAclFromScope(
        @SiteId                        uniqueidentifier,
        @ScopeId                       uniqueidentifier,
        @RequestGuid                   uniqueidentifier = NULL    OUTPUT
    );

    @SiteId: The Site Collection Identifier for the site collection containing the scope.
    @ScopeId: The Scope Identifier for the scope for which the Access Control List and anonymous User permissions are requested.
    @RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetAclFromScope stored procedure returns an integer return code which MUST be 0.
The **proc_SecGetAclFromScope** stored procedure MUST return a single result set as defined as follows.

### 3.1.5.62.1 ACL and Permission Result Set

The **ACL and Permission Result Set** (section 2.2.4.1) contains permission information for the specified scope. If either of the required input parameters is not valid, then the **ACL and Permission Result Set** MUST contain zero rows; otherwise, one row MUST be returned. The **ACL and Permission Result Set** is defined in Common Result Sets **ACL and Permission Result Set** section.

### 3.1.5.63 proc_SecGetAllAclsForSite

The **proc_SecGetAllAclsForSite** stored procedure is invoked to list all Scope Identifiers and their associated ACL and anonymous permission masks in a site collection.

```
PROCEDURE proc_SecGetAllAclsForSite(
  @SiteId                        uniqueidentifier,
  @MaxCount                      int,
  @RowCount                      int   out,
  @RequestGuid                   uniqueidentifier = NULL    OUTPUT
);
```

**@SiteId**: The Site Collection Identifier of the site collection for which the information is requested.

**@MaxCount**: The maximum size, in rows, allowed in the result set. If the number of unique scopes within the site collection exceeds **@MaxCount**, the **Access Control List Result Set** will not be returned.

**@RowCount**: Output parameter indicating the number of unique scopes that exist in the site collection.

**@RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_SecGetAllAclsForSite** stored procedure MUST return an integer return code of 0.

If the total count of unique Access Control Lists in the site collection does not exceed the value of the input parameter **@MaxCount**, then the **proc_SecGetAllAclsForSite** stored procedure MUST return the **Access Control List Result Set**. Otherwise, 0 result sets MUST be returned.

#### 3.1.5.63.1 Access Control List Result Set

The **Access Control List Result Set** returns one row for each unique scope and associated Access Control List defined in the site collection.

```
ScopeId                        uniqueidentifier,
Acl                            varbinary(max),
AnonymousPermMask              bigint;
```

**ScopeId**: The Scope Identifier of the scope.

**Acl**: The Access Control List (ACL) associated with the scope.

**AnonymousPermMask**: The **WSS Rights Mask** for anonymous users for this scope.
3.1.5.64 proc_SecGetAllGroupsAndMembershipInfo

The proc_SecGetAllGroupsAndMembershipInfo stored procedure is invoked to return information about all site groups and site group members within a site collection.

PROCEDURE proc_SecGetAllGroupsAndMembershipInfo(
  @SiteId uniqueidentifier,
  @RequestGuid uniqueidentifier = NULL OUTPUT
);

@SiteID: The Site Collection Identifier of the site collection to find all site groups in.

@RequestGuid: The optional request identifier for the current request.

Return Code Values: The proc_SecGetAllGroupsAndMembershipInfo stored procedure returns an integer which MUST be 0.

The proc_SecGetAllGroupsAndMembershipInfo stored procedure MUST return two result sets in the following order.

3.1.5.64.1 Groups Result Set

The Groups Result Set returns all site groups within the site collection, one row per site group, sorted by Site Group Identifier, in ascending order. The Groups Result Set MUST be empty if there is no site group on the site collection.

<table>
<thead>
<tr>
<th>ID</th>
<th>int,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>Description</td>
<td>nvarchar(512),</td>
</tr>
<tr>
<td>Owner</td>
<td>int,</td>
</tr>
<tr>
<td>OwnerIsUser</td>
<td>bit,</td>
</tr>
</tbody>
</table>

ID: The Site Group Identifier of the site group.

Title: The title of the site group.

Description: The description of the site group.

Owner: The User Identifier or Site Group Identifier of the owner of the site group.

OwnerIsUser: A bit value indicating whether the owner of the group is a User or a site group. If the owner is a User, this parameter’s value MUST be 1, otherwise, its value MUST be 0.

3.1.5.64.2 Group Membership Result Set

The Group Membership Result Set returns information about site group members who are not marked as deleted. A member can be a User or a domain group. Every row represents one site group member. The first column is the GroupId, the Site Group Identifier, followed by columns about the site group member. The Group Membership Result Set is sorted by GroupId, in ascending order. If a member belongs to more than one site group, then the member's information will appear in multiple rows, with different GroupIds. The Group Membership Result Set MUST be empty if none of the site groups have any members.

<table>
<thead>
<tr>
<th>GroupId</th>
<th>int,</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp_Id</td>
<td>int,</td>
</tr>
<tr>
<td>tp_SystemID</td>
<td>varbinary(512),</td>
</tr>
<tr>
<td>tp_Title</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>tp_Login</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>tp_Email</td>
<td>nvarchar(255),</td>
</tr>
</tbody>
</table>
tp_Notes                       nvarchar(1023),
tp_SiteAdmin                   bit,
tp_DomainGroup                 bit,
 tp_Flags                       int;

GroupId: The Site Group Identifier of the site group which contains this member.

tp_Id: The user identifier of the member who belongs to the site group specified by the GroupId in the first column.

tp_SystemID: The SystemID of the member.

tp_Title: The display name of the member.

tp_Login: The login name of the member.

tp_Email: The email address of the member. This parameter can be empty, but it MUST NOT be NULL.

tp_Notes: A string which contains extra information about the member. It can be empty, but it MUST NOT be NULL.

tp_SiteAdmin: A bit value indicating whether the member is a site collection Administrator. If the member is a site collection Administrator, this parameter's value MUST be 1, otherwise, it MUST be 0.

tp_DomainGroup: A bit value indicating whether the member is a domain group. If the site group member is a domain group, this parameter's value MUST be 1. Otherwise, its value MUST be 0.

tp_Flags: A 4-byte integer bit mask determining the user's options. See section 2.2.2.12.

3.1.5.65 proc_SecGetApplicationPrincipalAndUserToken

The proc_SecGetApplicationPrincipalAndUserToken stored procedure is invoked to return information about a principal (1) based on the principal's login name and user identifier.

PROCEDURE proc_SecGetApplicationPrincipalAndUserToken (  
    @SiteId                        uniqueidentifier,  
    @AppLogin                      nvarchar(255),  
    @UserId                        int,  
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
    );

@SiteId: The Site Collection Identifier (section 2.2.1.1.9) of the site collection containing the principal.

@AppLogin: The principal's login name.

@UserId: The user identifier of the principal.

@RequestGuid: The optional request identifier for the current request.

Return values: The proc_SecGetApplicationPrincipalAndUserToken stored procedure returns an integer return code, which MUST be 0.

If @UserId is NULL, then the proc_SecGetApplicationPrincipalAndUserToken stored procedure MUST return one result set. If @UserId is not NULL, the proc_SecGetApplicationPrincipalAndUserToken stored procedure MUST return two result sets in the order specified.

3.1.5.65.1 Application Principal Result Set
The **Application Principal Result Set** returns information on a principal (1) specified by @AppLogin. The **proc_SecGetApplicationPrincipalAndUserToken** stored procedure MUST return the Application Principal Result Set.

If @AppLogin does not match a principal, the Application Principal Result Set MUST be returned with zero rows. Otherwise, the Application Principal Result Set MUST be returned with a single row of data for the principal.

tp_ID: The user identifier of the principal.

tp_IsActive: Set to "1" if the current user is an active user in the site collection.

tp_Title: The principal's display name.

tp_Email: The principal's e-mail address.

tp_Token: A WSS User Token (section 2.2.3.10) value specifying the site group membership of the principal.

tp_Flags: Contains the user information flags value associated with this principal.

### 3.1.5.65.2 External Token Result Set

The External Token Result Set returns information on a principal (1) specified by @UserId. If @UserId is NULL, then the **proc_SecGetApplicationPrincipalAndUserToken** stored procedure MUST NOT return the External Token Result Set. If @UserId does not match a principal, then the External Token Result Set MUST be returned with zero rows. Otherwise, the External Token Result Set MUST be returned with a single row of data for the principal.

tp_ExternalToken: An External Group Token (section 2.2.3.2) value encoding information on external group membership derived from an external authentication role provider.

### 3.1.5.66 proc_SecGetCompleteWebRoleMemberList

The **proc_SecGetCompleteWebRoleMemberList** stored procedure is invoked to list all role assignments for all the principals with permissions on a specified Site.

```sql
PROCEDURE proc_SecGetCompleteWebRoleMemberList(
    @SiteId uniqueidentifier,
    @WebId uniqueidentifier,
    @LatestSecurityVersion bigint,
    @RequestGuid uniqueidentifier = NULL
    OUTPUT
);```

@SiteId: Site Collection Identifier of the site collection containing the Site whose permission information is requested.
@WebId: Site Identifier of the Site whose permission information is requested.

@LatestSecurityVersion: The current security version value for the specified site collection.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetCompleteWebRoleMemberList stored procedure returns an integer return code which MUST be 0.

The proc_SecGetCompleteWebRoleMemberList stored procedure MUST return one result set.

### 3.1.5.66.1 Role Member Result Set

The Role Member Result Set returns one row for each principal's role assignment for the specified Site. If the @SiteId or @WebId parameters are not valid, then zero rows MUST be returned.

#### RoleId

The Role Identifier of the role for the role assignment.

#### tp_Id

The site group identifier or user identifier of the principal (1).

#### tp_SystemID

The SystemId of the principal.

#### tp_DomainGroup

Set to 1 if the principal is a domain group; otherwise, 0.

### 3.1.5.67 proc_SecGetCurrentUsersCount

The proc_SecGetCurrentUsersCount stored procedure returns a result set containing a count of Users in the specified site collection.

```sql
PROCEDURE proc_SecGetCurrentUsersCount(
    @SiteId                        uniqueidentifier,
    @RequestGuid                   uniqueidentifier = NULL    OUTPUT
);
```

@SiteId: The Site Collection Identifier of the site collection.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetCurrentUsersCount stored procedure MUST return an integer return code of 0.

The proc_SecGetCurrentUsersCount stored procedure MUST return a single result set for the specified site collection as follows.

#### UsersCount

The number of Users registered with this site collection when configured in Active Directory account creation mode. When not in Active Directory account creation mode, UsersCount will be 1. The User Count Result Set MUST return one row for a given valid @SiteId, otherwise if @SiteId is invalid, zero rows MUST be returned.

UserQuota: int,
UsersCount: Contains the integer value for the number of Users registered with the specified site collection when configured in Active Directory account creation mode. In any other configuration, UsersCount MUST return 1.

UserQuota: Contains the limit for the number of Users allowed in the specified site collection. A value of 0 specifies no limit on the number of allowed Users.

{StorageQuotaError}: An error number generated when a site collection is over quota or locked. Its value MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Default value, no error.</td>
</tr>
<tr>
<td>212</td>
<td>The site collection is locked.</td>
</tr>
<tr>
<td>1816</td>
<td>Quota has been exceeded on this site collection.</td>
</tr>
</tbody>
</table>

3.1.5.68  proc_SecGetDomainGroupMapData

The proc_SecGetDomainGroupMapData stored procedure is invoked to retrieve the domain group map cache information for a site collection.

```sql
PROCEDURE proc_SecGetDomainGroupMapData(
    @SiteId                        uniqueidentifier,
    @DGCacheVersion                bigint,
    @RequestGuid                   uniqueidentifier = NULL
) OUTPUT;
```

@SiteId: The Site Collection Identifier.

@DGCacheVersion: The version number of the Domain Group Map Cache in the front-end Web server.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetDomainGroupMapData stored procedure returns an integer return code which MUST be 0.

The proc_SecGetDomainGroupMapData stored procedure MUST return 1 or 2 of the following result sets.

3.1.5.68.1  Domain Group Cache Versions Result Set

The Domain Group Cache Versions Result Set MUST be returned and MUST contain one row of version number data. If the specified @DGCacheVersion value is -2, then all columns returned MUST have the value -2, indicating that the value MUST NOT be used for comparison.

The Domain Group Cache Versions Result Set is defined in the Common Result Sets Domain Group Cache Versions Result Set (section 2.2.4.4).

3.1.5.68.2  Domain Group Cache BEDS Update Result Set
The **Domain Group Cache BEDS Update Result Set** MUST be returned if \( @DGCacheVersion \) does not equal "-2" and the value of **RealVersion** is greater than the value of **CachedVersion** in the results of the **Domain Group Cache Versions Result Set** (section 2.2.4.4). Otherwise, the **Domain Group Cache BEDS Update Result Set** MUST NOT be returned.

If the **Domain Group Cache BEDS Update Result Set** is returned, then there MUST be one row in the **Domain Group Cache BEDS Update Result Set** for each domain group, which is a member of a site group in the site collection, ordered by the identifier of the domain groups. A row is also returned for every other domain group that does not have a **GroupId** tied to it, including domain groups with NULL GroupIds.

The **Domain Group Cache BEDS Update Result Set** is defined in the Common Result Sets **Domain Group Cache BEDS Update Result Set** (section 2.2.4.3).

### 3.1.5.68.3 Domain Group Cache WFE Update Result Set

The **Domain Group Cache WFE Update Result Set** MUST be returned if \( @DGCacheVersion \) does not equal "-2" and the value of **RealVersion** is less than or equal to the value of **CachedVersion** in the results of the **Domain Group Cache Versions Result Set** (section 2.2.4.4). Otherwise, the **Domain Group Cache WFE Update Result Set** MUST NOT be returned.

If the **Domain Group Cache WFE Update Result Set** is returned, then it MUST contain one row.

The **Domain Group Cache WFE Update Result Set** is defined in the Common Result Sets **Domain Group Cache WFE Update Result Set** (section 2.2.4.5).

### 3.1.5.69 proc_SecGetGroupById

The **proc_SecGetGroupById** stored procedure is invoked to check whether the specified Group (either a site group or a domain group) exists in the specified site collection.

```sql
PROCEDURE proc_SecGetGroupById(
    @SiteId uniqueidentifier,
    @GroupId int,
    @Count int OUTPUT,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The **Site Collection Identifier** of the site collection to search for the Group.

@GroupId: The identifier (a **Site Group Identifier** for site groups, or a **User Identifier** for domain groups) of the specified Group.

@Count: Specifies whether the specified Group exists in the site collection as either a site group or a domain group. If the specified Group exists, @Count MUST be 1; otherwise, @Count MUST be 0.

@RequestGuid: The optional request identifier for the current request.

Return Code Values: The **proc_SecGetGroupById** stored procedure returns an integer which MUST be 0.

The **proc_SecGetGroupById** stored procedure MUST NOT return any result sets.

### 3.1.5.70 proc_SecGetGroupOwner

The **proc_SecGetGroupOwner** stored procedure is invoked to retrieve the User Identifier or Site Group Identifier for the owner of a site group. User Identifier and Group Identifier do not collide within the same site collection.
PROCEDURE proc_SecGetGroupOwner(
    @SiteId uniqueidentifier,
    @GroupId int,
    @OwnerId int OUTPUT,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);  

@SiteId: Site Collection Identifier for the site collection containing the site group.

@GroupId: The site group's identifier.

@OwnerId: An output parameter containing the User Identifier or Site Group Identifier of the site
group owner, or "-1" if the specified site group does not exist.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetGroupOwner stored procedure returns an integer return code which
MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>1319</td>
<td>The site group does not exist.</td>
</tr>
</tbody>
</table>

The proc_SecGetGroupOwner stored procedure MUST NOT return a result set.

3.1.5.71 proc_SecGetGroupSecurityScopes

The proc_SecGetGroupSecurityScopes stored procedure is invoked to retrieve a site group's role
assignments information on all security scopes within a given site collection.

PROCEDURE proc_SecGetGroupSecurityScopes(
    @SiteId uniqueidentifier,
    @PrincipalId int,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);  

@SiteId: The Site Collection Identifier of the site collection containing the site group.

@PrincipalId: The site group identifier.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetGroupSecurityScopes stored procedure MUST return an integer
return code of 0.

The proc_SecGetGroupSecurityScopes stored procedure MUST return the following result set.

3.1.5.71.1 Security Scopes Result Set

The Security Scopes Result Set returns scope information for the specified security principal. The
Security Scopes Result Set will be returned with zero or more rows.

ScopeUrl nvarchar(260),
Title    nvarchar(255);
**ScopeUrl**: The URL to the root of the Security Scope.

**Title**: The name of the role to which the site group is assigned on the Security Scope specified by `{ScopeUrl}`.

### 3.1.5.72 proc_SecGetIndividualUrlSecurityCheckEventReceivers

The `proc_SecGetIndividualUrlSecurityCheckEventReceivers` stored procedure is invoked to request security information and event receivers information about a document at a specified location. If the document does not exist, `proc_SecGetIndividualUrlSecurityCheckEventReceivers` provides security information about the specified location.

```sql
PROCEDURE proc_SecGetIndividualUrlSecurityCheckEventReceivers(
  @SiteId uniqueidentifier,
  @WebId uniqueidentifier,
  @FullUrl nvarchar(260),
  @DirName nvarchar(256),
  @LeafName nvarchar(128),
  @UserId int,
  @AttachmentsFlag tinyint,
  @bGetAttachmentWritePerm bit,
  @bGetListMetaData bit = 0,
  @bGetListScopes bit = 0,
  @Level tinyint = NULL,
  @HasEventReceiver bit OUTPUT,
  @MinLevel tinyint OUTPUT,
  @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

**@SiteId**: The site collection identifier of the site collection containing the document location.

**@WebId**: The site identifier of the site (2) containing the document location.

**@FullUrl**: The store-relative form URL of the document location, used to find a containing list or document library if the document does not exist.

**@DirName**: The directory name of the document location.

**@LeafName**: The leaf name of the document location.

**@UserId**: The user identifier of the current user, used to check for access privileges.

**@AttachmentsFlag**: An Attachments Flag (section 2.2.1.2.1) value specifying whether the document location is, or is contained within, an attachments folder.

**@bGetAttachmentWritePerm**: A bit flag specifying whether or not to return information about the write permissions of the current user to save an attachment at the specified document location. If this parameter is set to 1, and the `@AttachmentsFlag` parameter is not 0 or NULL, then the stored procedure MUST return the information about the current user's permissions as part of the Individual URL Security Result Set (section 3.1.5.72.1). This parameter MUST be ignored if `@AttachmentsFlag` is 0 or NULL.

**@bGetListMetaData**: A bit flag specifying whether or not metadata for the list or document library containing the specified document is requested.

**@bGetListScopes**: A bit flag specifying whether or not security scope information is requested for the list or document library containing the document location.

**@Level**: If not set to NULL, then this parameter specifies the Publishing Level Type (section 2.2.2.6) value to return in the Individual URL Security Result Set. If this parameter is NULL, the publishing level of the current version of the document for the current user MUST be used.
@HasEventReceiver: An output parameter that MUST be set to 1 if at least one event receiver is registered for the specified document; otherwise it is set to 0.

@MinLevel: A Publishing Level Type value that indicates the lowest value of publishing levels present for the document, returned as an output parameter.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetIndividualUrlSecurityCheckEventReceivers stored procedure returns an integer return code which MUST be 0.

The proc_SecGetIndividualUrlSecurityCheckEventReceivers stored procedure MUST return between 1 and 5 result sets according to the input parameters. See result set descriptions for more details. The result sets are returned in the following listed order.

3.1.5.72.1 Individual URL Security Result Set

The Individual URL Security Result Set contains security information about the specified document. If the document does not exist, but the specified URL is within a list or document library, security information is returned for the specified document location.

The Individual URL Security Result Set MUST be returned if the specified document location is contained within a list or document library. Otherwise, the NULL Individual URL Security Result Set (section 3.1.5.72.2) MUST be returned instead. If returned, the Individual URL Security Result Set MUST contain a single row.

The Individual URL Security Result Set is defined in the Common Result Sets Individual URL Security Result Set (section 2.2.4.12).

3.1.5.72.2 NULL Individual URL Security Result Set

The NULL Individual URL Security Result Set indicates that the document location is not contained within a List or document library. The NULL Individual URL Security Result Set MUST only be returned if the specified document location is not contained within a List or document library.

The NULL Individual URL Security Result Set is defined in the Common Result Sets NULL Individual URL Security Result Set (section 2.2.4.16).

3.1.5.72.3 List Metadata Result Set

The List Metadata Result Set returns the metadata for the list or document library containing the specified document location. The List Metadata Result Set MUST only be returned if the @bGetListMetadata parameter is set to 1 and the document location is contained within a list or document library, otherwise, the List Metadata Result Set is not returned. If returned, then the List Metadata Result Set MUST contain a single row.

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.4.14).

3.1.5.72.4 Event Receivers Result Set

The Event Receivers Result Set contains information about event receivers defined for the containing list or document library. This result set MUST only be returned if the @bGetListMetaData parameter is set to 1 and the document location is contained within a list or document library, otherwise, the Event Receivers Result Set is not returned. If returned, then the Event Receivers Result Set MUST contain one row for each event receiver registered for the list or document library.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.4.11).
3.1.5.72.5 List Scopes Result Set

The **List Scopes Result Set** returns the security information for the security scope associated with the document location. The **List Scopes Result Set** MUST only be returned if the `@bGetListScopes` parameter is set to 1 and the document location is within a list or document library, otherwise the **List Scopes Result Set** is not returned. If returned, then the **List Scopes Result Set** MUST contain one row.

```
ScopeId                        uniqueidentifier,
Acl                            varbinary(max),
AnonymousPermMask              bigint;
```

**ScopeId**: The scope identifier for the scope associated with the document location.

**Acl**: The ACL in **WSS ACL Format** (section 2.2.3.6) of the scope associated with the document location.

**AnonymousPermMask**: The **WSS Rights Mask** (section 2.2.2.15) that applies to an anonymous user, or a user with no assigned rights, in the scope associated with the document location.

3.1.5.73 proc_SecGetItemsWithUniquePermissions

The **proc_SecGetItemsWithUniquePermissions** stored procedure is invoked to return information about list items with unique permissions.

```
PROCEDURE proc_SecGetItemsWithUniquePermissions (   
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @ListId                        uniqueidentifier,
    @TitleMode                     int,
    @MaxItemToReturn               int,
    @FolderOnly                    bit,
    @RequestGuid                   uniqueidentifier   = NULL OUTPUT   
);   
```

**@SiteId**: The Site Collection Identifier of the site collection containing the list items.

**@WebId**: The Site Identifier of the site (2) containing the list items.

**@ListId**: The List Identifier of the list containing the list items.

**@TitleMode**: This determines how title information is returned. If the list is a document library, then **@TitleMode** MUST be 0. If the list has a **List Base Type** (section 2.2.1.2.11) of issues list, or a **List Server Template** (section 2.2.1.2.12) of links list, then **@TitleMode** MUST be 1. Otherwise, **@TitleMode** MUST be 2.

**@MaxItemToReturn**: The maximum number of rows to be returned in the result set.

**@FolderOnly**: If this is not 0, then the result set MUST contain only information about folders. Otherwise, information about both folders and non-folders will be returned.

**@RequestGuid**: The optional request identifier for the current request.

Return values: The **proc_SecGetItemsWithUniquePermissions** stored procedure returns an integer return code, which MUST be 0.

This **proc_SecGetItemsWithUniquePermissions** stored procedure MUST return one result set.
3.1.5.73.1 Items with Unique Permissions Result Set

The Items with Unique Permissions Result Set returns one row for each list item with unique permissions.

ScopeUrl: The URL to the root of the security scope.

DoclibRowId: The row identifier for the document within the list.

{Title}: If @TitleMode is 0, then {Title} MUST be the leaf name of the document. If @TitleMode is 1, then {Title} MUST be the list item identifier. If @TitleMode is 2, then {Title} MUST contain the title of this list item.

IsFolder: If this list item is a folder, IsFolder MUST be 1. Otherwise, IsFolder MUST be 0.

3.1.5.74 proc_SecGetPrincipalByEmail

The proc_SecGetPrincipalByEmail stored procedure is invoked to return user information about a user associated with a specified email address.

PROCEDURE proc_SecGetPrincipalByEmail(  
   @SiteId                        uniqueidentifier,  
   @Email                         nvarchar(255),  
   @RequestGuid                   uniqueidentifier= NULL OUTPUT  
);

@SiteId: The site collection identifier for the site collection containing the user.

@Email: The user's email address. If this parameter is NULL, the email address MUST be the empty string.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetPrincipalByEmail stored procedure returns an integer return code, which MUST be 0.

The proc_SecGetPrincipalByEmail stored procedure MUST return a single result set.

3.1.5.74.1 Principal User Information Result Set

The Principal User Information Result Set returns information about the User associated with the specified email address. The Principal User Information Result Set MUST contain zero rows if no users have the specified email address or if the associated user has been marked as deleted. The Principal User Information Result Set is defined in the Common Result Sets Principal User Information Result Set (section 2.2.4.19).

3.1.5.75 proc_SecGetPrincipalById

The proc_SecGetPrincipalById stored procedure is invoked to return information about a principal (1) or collection of principals based on a specified site group identifier or user identifier.

PROCEDURE proc_SecGetPrincipalById(  

@SiteId: The site collection identifier of the site collection containing the principal.

@PrincipalId: The site group identifier of the site group or user identifier of the principal to return information for.

@GetSTSToken: If this parameter is set to 1, then it indicates the tp_token value MUST be returned in both instances of the Principal User Information Result Set.

@GetExternalToken: If this parameter is set to 1, then it indicates the tp_ExternalTokenLastUpdated and tp_ExternalToken values MUST be returned in both instances of the Principal User Information Result Set.

@GetExpandedSTSGroup: If this parameter is set to 1, then it indicates a second Principal User Information Result Set MUST be returned when the first instance of the Principal User Information Result Set has zero rows. The second result set contains information about group membership for the site group represented by @PrincipalId. The rows returned in the second result set are all the members of that site group.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetPrincipalById stored procedure returns an integer return code, which MUST be 0.

The proc_SecGetPrincipalById stored procedure MUST return one result set which can be returned either once or twice as described in the following.

### 3.1.5.75.1 User Information Result Set

The User Information Result Set returns information on a specified principal or the members of a site group.

If @PrincipalId matches a principal, the User Information Result Set MUST be returned once with a single row of data for the principal.

Otherwise, the User Information Result Set MUST be returned with zero rows, and if @GetExpandedSTSGroup is 1, another instance of the User Information Result Set MUST be returned, with one row of data for each principal within the site group.
tp_ID: The user identifier of the principal.

tp_SystemID: The principal's SystemId (section 2.2.1.12).

tp_Title: The principal's display name.

tp_Login: The principal's login name.

tp_Email: The principal's e-mail address.

tp_Notes: A longer descriptive text associated with the principal.

tp_SiteAdmin: Set to 1 if the principal is a site collection administrator; otherwise, 0.

 tp_SiteAdmin: The principal's SystemId (section 2.2.1.12).

{tp_SiteAdmin}: Set to 1 if the principal is a domain group; otherwise, 0.

{tp_SiteAdmin}: Contains the user information flags value describing this principal.

{tp_ExternalTokenLastUpdated}: A timestamp in UTC format specifying the time when the External Group Token (section 2.2.3.2) was last updated. This parameter MUST be NULL if @GetExternalToken is 0.

{tp_ExternalToken}: An External Group Token value specifying the domain group membership of the principal. This parameter MUST be NULL if @GetExternalToken is 0.

{tp_Token}: A WSS User Token (section 2.2.3.10) value specifying the site group membership of the principal. This parameter MUST be NULL if @GetSTSToken is 0.

3.1.5.76 proc_SecGetPrincipalByIdEx

The proc_SecGetPrincipalByIdEx stored procedure is invoked to return information about a principal based on a specified user identifier or a collection of principals based on a specified site group identifier. The proc_SecGetPrincipalByIdEx stored procedure is similar to the proc_SecGetPrincipalById stored procedure, except that the proc_SecGetPrincipalByIdEx stored procedure also returns the principal's mobile phone information.

PROCEDURE proc_SecGetPrincipalByIdEx (
@SiteId                        uniqueidentifier,
@PrincipalId                   int,
@GetSTSToken                   bit                = 0,
@GetExternalToken              bit                = 0,
@GetExpandedSTSGroup           bit                = 0,
@RequestGuid                   uniqueidentifier   = NULL OUTPUT
)

@SiteId: The Site Collection Identifier of the site collection containing the principal(s).

@PrincipalId: The site group identifier of the site group or user identifier of the principal(s) to return information for.

@GetSTSToken: If this parameter is set to 1, it indicates the tp_token value MUST be returned in both instances of the Principal User Information Result Set.

@GetExternalToken: If this parameter is set to 1, it indicates the tp_ExternalTokenLastUpdated and tp_ExternalToken values MUST be returned in both instances of the Principal User Information Result Set.

@GetExpandedSTSGroup: If this parameter is set to 1, it indicates a second Principal User Information Result Set MUST be returned when the first instance of the Principal User Information Result Set has zero rows. The second result set contains information about group...
membership for the site group represented by @PrincipalId. The rows returned in the second result set are all the members of that site group.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetPrincipalById stored procedure returns an integer return code, which MUST be 0.

The proc_SecGetPrincipalByIdEx stored procedure MUST return one result set which can be returned either once or twice as described in the following.

### 3.1.5.76.1 Extended Principal User Information Result Set

The **Extended Principal User Information Result Set** returns information on a specified principal (1) or the principals that are members of a site group.

If @PrincipalId matches a principal, then the **Extended Principal User Information Result Set** MUST be returned once with a single row of data for the principal.

Otherwise, the **Extended Principal User Information Result Set** MUST be returned with zero rows, and if @GetExpandedSTSGroup is 1, another instance of the **Extended Principal User Information Result Set** MUST be returned, with one row of data for each principal within the site group.

```plaintext
<table>
<thead>
<tr>
<th>tp_ID</th>
<th>int</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp_SystemID</td>
<td>varbinary(512)</td>
</tr>
<tr>
<td>tp_Title</td>
<td>nvarchar(255)</td>
</tr>
<tr>
<td>tp_Login</td>
<td>nvarchar(255)</td>
</tr>
<tr>
<td>tp_Email</td>
<td>nvarchar(255)</td>
</tr>
<tr>
<td>tp_Notes</td>
<td>nvarchar(1023)</td>
</tr>
<tr>
<td>tp_SiteAdmin</td>
<td>bit</td>
</tr>
<tr>
<td>tp_DomainGroup</td>
<td>bit</td>
</tr>
<tr>
<td>tp_Flags</td>
<td>int</td>
</tr>
<tr>
<td>{tp_ExternalTokenLastUpdated}</td>
<td>datetime</td>
</tr>
<tr>
<td>{tp_ExternalToken}</td>
<td>varbinary(max)</td>
</tr>
<tr>
<td>{tp_Token}</td>
<td>varbinary(max)</td>
</tr>
<tr>
<td>tp_Mobile</td>
<td>nvarchar(127)</td>
</tr>
</tbody>
</table>
```

**tp_ID**: The user identifier of the principal.

**tp_SystemID**: The principal's SystemId (section 2.2.1.12).

**tp_Title**: The principal's display name.

**tp_Login**: The principal's login name.

**tp_Email**: The principal's e-mail address.

**tp_Notes**: A longer descriptive text associated with the principal.

**tp_SiteAdmin**: Set to 1 if the principal is a site collection administrator; otherwise, 0.

**tp_DomainGroup**: Set to 1 if the principal is a domain group; otherwise, 0.

**tp_Flags**: Contains the User Information Flags value describing this principal.

**{tp_ExternalTokenLastUpdated}**: A timestamp in UTC format specifying the time when the External Group Token (section 2.2.3.2) was last updated. This parameter MUST be NULL if @GetExternalToken is 0.

**{tp_ExternalToken}**: An External Group Token value specifying the domain group membership of the principal. This parameter MUST be NULL if @GetExternalToken is 0.
3.1.5.77 proc_SecGetPrincipalByLogin

The proc_SecGetPrincipalByLogin stored procedure is invoked to return security and attribute information for a principal (a user or domain group) identified by a specified login name.

PROCEDURE proc_SecGetPrincipalByLogin(
    @SiteId                        uniqueidentifier,
    @Login                         nvarchar(255),
    @GetSTSToken                   bit                = 0,
    @GetExternalToken              bit                = 0,
    @RequestGuid                   uniqueidentifier   = NULL OUTPUT
);

@SiteId: The Site Collection Identifier for the site collection associated with the principal whose information is requested.

@Login: The login name of the principal. This parameter MUST NOT be NULL.

@GetSTSToken: A bit flag specifying whether to include site group membership information for the principal. If this parameter is set to 1, then a WSS User Token (section 2.2.3.10) containing the principal’s site group membership information MUST be returned in the User Information Result Set.

@GetExternalToken: A bit flag specifying whether to include domain group membership and related timestamp information for the principal. If this parameter is set to 1, then an External Group Token (section 2.2.3.2) containing the principal’s domain group membership information and a timestamp indicating the principal’s most recent update time MUST be returned in the User Information Result Set.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetPrincipalByLogin stored procedure returns an integer return code, which MUST be 0.

The proc_SecGetPrincipalByLogin stored procedure MUST return a single result set as follows.

3.1.5.77.1 User Information Result Set

The User Information Result Set returns security and attribute information about a specified principal (1). The User Information Result Set MUST be returned, and MUST contain one row if the @Login parameter matches an existing principal who is not marked as deleted in the UserInfo Table (section 2.2.6.10); otherwise it MUST contain no rows.

tp_ID                          int,
tp_SystemID                    varbinary(512),
tp_Title                       nvarchar(255),
tp_Login                       nvarchar(255),
tp_Email                       nvarchar(255),
tp_Notes                       nvarchar(1023),
tp_SiteAdmin                   bit,
tp_DomainGroup                 bit,
tp_Flags                       bit,
{tp_ExternalTokenLastUpdated}  datetime,
{tp_ExternalToken}             varbinary(max),

{tp_Token}: A WSS User Token (section 2.2.3.10) value specifying the site group membership of the principal. This parameter MUST be NULL if @GetSTSToken is 0.

tp_Mobile: The principal's mobile phone number.
**tp_ID**: The User Identifier for the specified principal.

**tp_SystemID**: The SystemId (section 2.2.1.1.12) for the principal.

**tp_Title**: The display name of the principal.

**tp_Login**: The principal's login name.

**tp_Email**: The principal's email address.

**tp_Notes**: A descriptive text string associated with the principal.

**tp_SiteAdmin**: A bit set to 1 if the principal has administrator rights in the site collection; otherwise, 0.

**tp_DomainGroup**: A bit set to 1 if the principal is a domain group; otherwise 0, indicating the principal is a user.

**tp_Flags**: Contains the **User Information Flags** value describing this principal.

**{tp_ExternalTokenLastUpdated}**: A datetime in UTC format specifying the time when the external group token was last updated. This value MUST be NULL if the @GetExternalToken parameter is not set to 1.

**{tp_ExternalToken}**: An External Group Token (section 2.2.3.2) value specifying the domain group membership of the principal. This value MUST be NULL if the @GetExternalToken parameter is not set to 1.

**{tp_Token}**: A WSS User Token value specifying the site group membership of the principal. This value MUST be NULL if the @GetSTSToken parameter is not set to 1.

### 3.1.5.78  proc_SecGetPrincipalByLogin20

The **proc_SecGetPrincipalByLogin20** stored procedure is invoked to return security principal information based on up to 20 separate login names.

```sql
PROCEDURE proc_SecGetPrincipalByLogin20(
    @SiteId uniqueidentifier,
    @PrincipalId01 nvarchar(255),
    @PrincipalId02 nvarchar(255),
    @PrincipalId03 nvarchar(255),
    @PrincipalId04 nvarchar(255),
    @PrincipalId05 nvarchar(255),
    @PrincipalId06 nvarchar(255),
    @PrincipalId07 nvarchar(255),
    @PrincipalId08 nvarchar(255),
    @PrincipalId09 nvarchar(255),
    @PrincipalId10 nvarchar(255),
    @PrincipalId11 nvarchar(255),
    @PrincipalId12 nvarchar(255),
    @PrincipalId13 nvarchar(255),
    @PrincipalId14 nvarchar(255),
    @PrincipalId15 nvarchar(255),
    @PrincipalId16 nvarchar(255),
    @PrincipalId17 nvarchar(255),
    @PrincipalId18 nvarchar(255),
    @PrincipalId19 nvarchar(255),
    @PrincipalId20 nvarchar(255),
    @RequestGuid uniqueidentifier = NULL OUTPUT
```
@SiteId: The Site Collection Identifier for the site collection that contains the security principals (2).

@PrincipalId##: The login names for Users to be returned.

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_SecGetPrincipalByLogin20` stored procedure returns an integer return code which MUST be 0.

The `proc_SecGetPrincipalByLogin20` stored procedure MUST return one result set of security principal information for each non-NULL @PrincipalId##:

3.1.5.78.1 User Information Result Set

The User Information Result Set returns User information on a specified security principal. The User Information Result Set MUST return one row if the @PrincipalId## matches a security principal's (2) ID, otherwise it MUST contain no rows. The User Information Result Set is defined in the `proc_SecGetPrincipalByLogin User Information Result Set` (section 3.1.5.19.5).

3.1.5.79 proc_SecGetPrincipalDisplayInformation20

The `proc_SecGetPrincipalDisplayInformation20` stored procedure is invoked to return security principal or site group information for up to 20 principal identifiers.

    PROCEDURE proc_SecGetPrincipalDisplayInformation20(
        @SiteId                        uniqueidentifier,
        @WebId                         uniqueidentifier,
        @PrincipalId01                 int,
        @PrincipalId02                 int,
        @PrincipalId03                 int,
        @PrincipalId04                 int,
        @PrincipalId05                 int,
        @PrincipalId06                 int,
        @PrincipalId07                 int,
        @PrincipalId08                 int,
        @PrincipalId09                 int,
        @PrincipalId10                 int,
        @PrincipalId11                 int,
        @PrincipalId12                 int,
        @PrincipalId13                 int,
        @PrincipalId14                 int,
        @PrincipalId15                 int,
        @PrincipalId16                 int,
        @PrincipalId17                 int,
        @PrincipalId18                 int,
        @PrincipalId19                 int,
        @PrincipalId20                 int,
        @RequestGuid                   uniqueidentifier = NULL OUTPUT
    );

@SiteId: The site collection identifier for the site collection containing the security principals (2) and the Security Groups to be listed.

@WebId: A site identifier for a site (2). This parameter is ignored.

@PrincipalId##: The identifier for a security principal or a site group to be returned. Result sets are returned for each non-NULL @PrincipalId## parameter.
@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetPrincipalDisplayInformation stored procedure returns an integer return code which MUST be 0.

The proc_SecGetPrincipalDisplayInformation stored procedure MUST return one or two result sets for each non-NUL @PrincipalId#

### 3.1.5.79.1 Site Group Principal Display Information Result Set

The Site Group Principal Display Information Result Set returns information on a specified @PrincipalId#. The @PrincipalId# MUST match the ID of either a site group or a principal (1). If @PrincipalId# matches the ID of a site group, then the Site Group Principal Display Information Result Set MUST return one row. If the @PrincipalId# matches the ID of a principal, then the Site Group Principal Display Information Result Set MUST have no rows.

```sql
IsUser bit,
IsSiteGroup bit,
UserID int,
UserSID varbinary(512),
UserName nvarchar(255),
UserLogin nvarchar(255),
UserNotes nvarchar(1023),
UserSiteAdmin bit,
UserDomainGroup bit,
GroupID int,
GroupName nvarchar(255),
GroupDescription nvarchar(512),
GroupOwnerID int,
GroupOwnerIsUser bit,
GroupType tinyint;
```

**IsUser**: MUST be 0.

**IsSiteGroup**: MUST be 1.

**UserID**: MUST be NULL.

**UserSID**: MUST be NULL.

**UserName**: MUST be NULL.

**UserLogin**: MUST be NULL.

**UserEmail**: MUST be NULL.

**UserNotes**: MUST be NULL.

**UserSiteAdmin**: MUST be NULL.

**UserDomainGroup**: MUST be NULL.

**GroupID**: The site group identifier. MUST match @PrincipalId#.

**GroupName**: The display name of the site group, specified by @GroupID.

**GroupDescription**: The description of the site group specified by @GroupID.

**GroupOwnerID**: The identifier of the user who owns the site group, specified by @GroupID.
**GroupOwnerIsUser**: A bit set to 1 if the user who owns of the site group is a principal; otherwise it MUST be 0.

**GroupType**: MUST be 0.

### 3.1.5.79.2 Security Principal Display Information Result Set

If @PrincipalId## matches the ID of a site group, then the **Security Principal Display Information Result Set** MUST NOT be returned. If the @PrincipalId## matches the ID of a **security principal**, then the **Security Principal Display Information Result Set** MUST have one row for the matching security principal.

IsUser: MUST be 1.

IsSiteGroup: MUST be 0.

UserID: The user identifier of the principal. MUST match @PrincipalId##.

UserSID: The **SystemID** of the principal.

UserName: The display name of the principal.

UserLogin: The login name of the principal.

UserEmail: The email address of the principal.

UserNotes: Description text associated with the principal.

UserSiteAdmin: Set to 1 to indicate that the principal is a site collection administrator, otherwise 0.

UserDomainGroup: Set to 1 to indicate that the principal is a domain group, otherwise 0.

UserFlags: Contains the user information flags value describing this principal.

GroupId: MUST be NULL.

GroupName: MUST be NULL.

GroupDescription: MUST be NULL.

GroupOwnerID: MUST be NULL.

GroupOwnerIsUser: MUST be NULL.
**GroupType:** MUST be NULL.

### 3.1.5.80 proc_SecGetRoleAssignments

The **proc_SecGetRoleAssignments** stored procedure is invoked to request the WSS ACE (section 2.2.3.5) for a specified security scope in a site collection.

```sql
PROCEDURE proc_SecGetRoleAssignments (  
    @SiteId                        uniqueidentifier,
    @ScopeId                       uniqueidentifier,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);  
```

@**SiteID**: The site collection identifier of the site collection containing the requested security scope.

@**ScopeID**: The scope identifier of the security scope containing the requested WSS ACEs.

@**RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_SecGetRoleAssignments** stored procedure returns an integer return code which MUST be 0.

The **proc_SecGetRoleAssignments** stored procedure MUST return one result set. It will return zero or more rows.

#### 3.1.5.80.1 WSSACE Result Set

The **WSSACE Result Set** MUST return one row for each role assignment matching the given security scope within the site collection. Each row represents a WSS ACE (section 2.2.3.5).

```sql
PrincipalId                    int,
PermMask                       bigint;
```

**PrincipalId**: A 4 byte signed integer specifying the user identifier of the principal for this WSS ACE.

**PermMask**: A WSS Rights Mask (section 2.2.2.15) that contains the list of rights that is granted to the principal.

### 3.1.5.81 proc_SecGetRoleBindingsForAllPrincipals

The **proc_SecGetRoleBindingsForAllPrincipals** stored procedure is invoked to list the role assignments for all principals in a security scope.

```sql
PROCEDURE proc_SecGetRoleBindingsForAllPrincipals(  
    @SiteId                        uniqueidentifier,
    @ScopeId                       uniqueidentifier,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);  
```

@**SiteId**: The site collection identifier of the site collection containing the security scope.

@**ScopeId**: The scope identifier of the security scope containing the role assignments.

@**RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_SecGetRoleBindingsForAllPrincipals** stored procedure MUST return an integer return code of 0.
The `proc_SecGetRoleBindingsForAll Principals` stored procedure MUST return one result set as follows.

### 3.1.5.81.1 Role Assignment Result Set

The **Role Assignment Result Set** MUST return one row for each undeleted principal in the scope.

<table>
<thead>
<tr>
<th>RoleId</th>
<th>int,</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrincipalId</td>
<td>int;</td>
</tr>
</tbody>
</table>

**RoleId**: The role identifier of a role associated with the role assignment.

**PrincipalId**: The user identifier of the principal.

### 3.1.5.82 proc_SecGetRoleDefs

`proc_SecGetRoleDefs` is invoked to retrieve role definition information for items in the specified site collection and scope.

```sql
PROCEDURE proc_SecGetRoleDefs(
    @SiteId uniqueidentifier,
    @ScopeId uniqueidentifier,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

**@SiteId**: The site collection identifier for the desired site collection. This parameter MUST correspond to a valid site collection.

**@ScopeId**: The scope identifier of the desired scope.

**@RequestGuid**: The optional request identifier for the current request.

Return Values: The `proc_SecGetRoleDefs` stored procedure MUST return an integer return code of 0, and MUST return a role definition result set.

### 3.1.5.82.1 Role Definition Result Set

The **Role Definition Result Set** contains a list of role definitions. The **Role Definition Result Set** MUST contain one row for each role defined in the specified site collection at the specified scope.

<table>
<thead>
<tr>
<th>RoleId</th>
<th>int,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>nvarchar(255),</td>
</tr>
<tr>
<td>Description</td>
<td>nvarchar(512),</td>
</tr>
<tr>
<td>Hidden</td>
<td>bit,</td>
</tr>
<tr>
<td>RoleOrder</td>
<td>int,</td>
</tr>
<tr>
<td>Type</td>
<td>tinyint,</td>
</tr>
<tr>
<td>WebId</td>
<td>uniqueidentifier,</td>
</tr>
<tr>
<td>{PermMaskUpper}</td>
<td>int,</td>
</tr>
<tr>
<td>{PermMaskLower}</td>
<td>int,</td>
</tr>
<tr>
<td>WebGroupId</td>
<td>int;</td>
</tr>
</tbody>
</table>

**RoleId**: The role identifier for the role.

**Title**: The display name of the role.

**Description**: A user-defined description of the role.
Hidden: Implementation-specific bit flag used to inform the front-end Web server whether or not to display the role definition.

RoleOrder: Specifies the order in which roles MUST be displayed in the front-end Web server. Roles MUST be sorted first in ascending RoleOrder, then by descending Type.

Type: The Role Definition Type (section 2.2.1.16) of the role.

WebId: The Site Identifier for the site (2) within the site collection to which the role is assigned.

{PermMaskUpper}: An integer containing the high-order 32 bits of a WSS Rights Mask (section 2.2.2.15) defining the permissions for the role.

{PermMaskLower}: An integer containing the low-order 32 bits of a WSS Rights Mask defining the permissions for the role.

WebGroupId: If the role was upgraded from a site group, then WebGroupId MUST be the identifier of the site group from which it was upgraded, otherwise, it MUST be -1.

3.1.5.83 proc_SecGetSecurityInfo

The proc_SecGetSecurityInfo stored procedure retrieves security permissions information about a document. The document can be specified using its document identifier or its URL.

```
PROCEDURE proc_SecGetSecurityInfo(
    @SiteId uniqueidentifier,
    @WebId uniqueidentifier,
    @Url nvarchar(260),
    @ThresholdCount int,
    @DocId uniqueidentifier,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the document. This parameter MUST be ignored.

@Url: The store-relative form URL for the document. The @Url parameter MUST be NULL to specify the document with the @DocId parameter.

@ThresholdCount: If @Url is a folder and the list (1) does not have more list items than @ThresholdCount, the BigListFolder field in the Security Information Result Set (section 3.1.5.83.1) MUST be zero.

@DocId: The document identifier of the document. The @DocId parameter MUST be ignored if @Url is not NULL.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetSecurityInfo stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The document was not found.</td>
</tr>
</tbody>
</table>
The **proc_SecGetSecurityInfo** stored procedure MUST return a single **Security Information Result Set** (section 3.1.5.83.1) upon successful execution, or no result sets if the document is not found.

### 3.1.5.83.1 Security Information Result Set

The **Security Information Result Set** returns information about the security permissions on the document. The **Security Information Result Set** MUST be returned only if the document exists, and MUST return one row of information.

```
{ScopeId}                        uniqueidentifier,  
{ScopeUrl}                       nvarchar(260),  
{IsScope}                      bit,  
Acl                            varbinary(max),  
AnonymousPermMask              bigint,  
{BigListFolder}                  bit;  
{ListId}                        uniqueidentifier
```

- **{ScopeId}**: Contains the **scope identifier** of the scope that contains the document.
- **{ScopeUrl}**: The store-relative form URL of the scope that contains the document.
- **{IsScope}**: A flag indicating that the document is the securable object at the root of the scope. This value MUST be set to 1 if ScopeUrl is the store-relative form URL of the document; otherwise, the value MUST be 0.
- **Acl**: The **access control list (ACL)** of the scope that contains the document.
- **AnonymousPermMask**: The **WSS Rights Mask** (section 2.2.2.15) in effect for anonymous users in the scope.
- **{BigListFolder}**: A flag indicating whether or not the list is considered too big for non-administrators. If @Url is not a folder, then **BigListFolder** MUST be 0. If the list is not throttled, **BigListFolder** MUST be 0. If the list does not contain more list items than @ThresholdCount, then **BigListFolder** MUST be 0. Otherwise, **BigListFolder** MUST be 1.
- **{ListId}**: The **list identifier** of the list containing the document.

### 3.1.5.84 proc_SecGetSiteAdmins

The **proc_SecGetSiteAdmins** stored procedure is invoked to return a list of all site collection administrators not marked as deleted.

```
PROCEDURE proc_SecGetSiteAdmins(
    @SiteId                   uniqueidentifier,  
    @RequestGuid             uniqueidentifier              OUTPUT
);  
```

- **@SiteId**: The site collection identifier of the site collection containing the site collection administrators.
- **@RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_SecGetSiteAdmins** stored procedure returns an integer return code, which MUST be 0 and it MUST return a single result set

### 3.1.5.84.1 Site Administrators Result Set
The **Site Administrators Result Set** returns a list of all site collection administrators not marked as deleted. The **Site Administrators Result Set** MUST be returned. The **Site Administrators Result Set** MUST return one row per site collection administrator.

```sql
  tp_Id                          int,
  tp_SystemID                    varbinary(512),
  tp_Title                       nvarchar(255),
  tp_Login                       nvarchar(255),
  tp_Email                       nvarchar(255),
  tp_Notes                       nvarchar(1023),
  tp_SiteAdmin                   bit,
  tp_DomainGroup                 bit,
  tp_Flags                       int;
```

**tp_Id**: The user identifier of the security principal (2) who is a site collection administrator.

**tp_SystemID**: The SystemID of the security principal (2).

**tp_Title**: The name of the security principal (2), used for display on the front-end Web server.

**tp_Login**: The login name of the security principal (2), for example, EXAMPLE\username.

**tp_Email**: The email address of the security principal (2), for example, "username@mail.example.com".

**tp_Notes**: Notes associated with the site collection administrator.

**tp_SiteAdmin**: This value is set to 1 if the security principal (2) is a site collection administrator. This value is set to 0 if the security principal (2) is not a site collection administrator.

**tp_DomainGroup**: This value is set to 1 if the security principal (2) is a domain group. This value is set to 0 if the security principal (2) is not a domain group.

**tp_Flags**: A 4-byte integer bit mask determining the security principal's (2) options as specified in **UserInfo Flags** (section 2.2.2.12).

### 3.1.5.85 proc_SecGetSiteGroupById

The **proc_SecGetSiteGroupById** stored procedure gets information about a site group given its ID.

```sql
PROCEDURE proc_SecGetSiteGroupById(
    @SiteId                        uniqueidentifier,
    @GroupId                       int,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

**@SiteId**: The Site Collection Identifier of the site collection containing the site group.

**@GroupId**: The identifier of the site group.

**@RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_SecGetSiteGroupById** stored procedure MUST return an integer return code of 0.

The **proc_SecGetSiteGroupById** stored procedure MUST return a single result set.

### 3.1.5.85.1 Site Group Result Set
The Site Group Result Set returns information about a site group. The Site Group Result Set MUST contain 1 row of site group information if the Sec_SiteGroupsView.SiteWebID equals @SiteId and Sec_SiteGroupsView.ID equals @GroupID; otherwise, 0 rows MUST be returned.

The Site Group Result Set is defined using T-SQL syntax, in the Sec_SiteGroupsView (section 2.2.6.5).

3.1.5.86 proc_SecGetSiteGroupByTitle

The proc_SecGetSiteGroupByTitle stored procedure is invoked to get site group information for the site group with the specified user-friendly name.

```
PROCEDURE proc_SecGetSiteGroupByTitle(
    @SiteId                        uniqueidentifier,
    @Title                         nvarchar(255),
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The Site Collection Identifier of the site collection containing the site group.

@Title: The user-friendly name of the site group.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetSiteGroupByTitle stored procedure MUST return an integer return code of 0 and it MUST return a single result set.

3.1.5.86.1 Site Group Information Result Set

The Site Group Information Result Set returns the available site group information for the specified site group. The Site Group Information Result Set MUST contain one row of site group information if the @SiteID parameter value and the @Title parameter value match an existing site group; otherwise, 0 rows MUST be returned.

The Site Group Information Result Set schema is defined in the Sec_SiteGroupsView (section 2.2.6.5).

3.1.5.87 proc_SecGetSiteGroupByTitle20

The proc_SecGetSiteGroupByTitle20 stored procedure provides information about site groups in bulk, as specified by a set of up to 20 site group titles.

```
PROCEDURE proc_SecGetSiteGroupByTitle20(
    @SiteId                        uniqueidentifier,
    @PrincipalId01                 nvarchar(255),
    @PrincipalId02                 nvarchar(255),
    @PrincipalId03                 nvarchar(255),
    @PrincipalId04                 nvarchar(255),
    @PrincipalId05                 nvarchar(255),
    @PrincipalId06                 nvarchar(255),
    @PrincipalId07                 nvarchar(255),
    @PrincipalId08                 nvarchar(255),
    @PrincipalId09                 nvarchar(255),
    @PrincipalId10                 nvarchar(255),
    @PrincipalId11                 nvarchar(255),
    @PrincipalId12                 nvarchar(255),
    @PrincipalId13                 nvarchar(255),
    @PrincipalId14                 nvarchar(255),
    @PrincipalId15                 nvarchar(255),
);
@PrincipalId16                 nvarchar(255),
@PrincipalId17                 nvarchar(255),
@PrincipalId18                 nvarchar(255),
@PrincipalId19                 nvarchar(255),
@PrincipalId20                 nvarchar(255),
@RequestGuid                   uniqueidentifier
);

@SiteId: The Site Collection Identifier for the site collection containing the specified site groups.

@PrincipalId##: Twenty Site Group Title fields (01 - 20).

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetSiteGroupByTitle20 stored procedure MUST return an integer return code of 0.

The proc_SecGetSiteGroupByTitle20 stored procedure MUST return one result set for each NON-NULL @PrincipalId##.

3.1.5.87.1 Site Group Information Result Set

The Site Group Information Result Set returns once per each NON-NULL @PrincipalId##, and the Site Group Information Result Set MUST contain one row of site group information if the @SiteID parameter value and the @PrincipalId## parameter value match an existing site group; otherwise, zero rows MUST be returned.

The Site Group Information Result Set is defined using T-SQL syntax, as defined by the Sec_SiteGroupsView (section 2.2.6.5).

3.1.5.88 proc_SecGetUserAccountDirectoryPath

The proc_SecGetUserAccountDirectoryPath stored procedure is invoked to return the User Account Directory Path of a specified site collection.

This stored procedure is called when a user has specific user restrictions and the User Account Directory Path is required along other user information.

PROCEDURE proc_SecGetUserAccountDirectoryPath(
    @SiteId                            uniqueidentifier,
    @RequestGuid                       uniqueidentifier          OUTPUT
);

@SiteId: The site collection identifier of the site collection.

@RequestGuid: The optional request identifier for the current request.

Return Code Values: The proc_SecGetUserAccountDirectoryPath stored procedure returns an integer return code, which MUST be 0.

The proc_SecGetUserAccountDirectoryPath stored procedure MUST return one result set.

3.1.5.88.1 User Account Directory Path Result Set

The User Account Directory Path Result Set returns the User Account Directory Path for the site collection. The User Account Directory Path Result Set MUST contain one row when the specified site collection exists, and it MUST contain zero rows when the specified site collection does not exist.
UserAccountDirectoryPath: Contains the User Account Directory Path for the site collection.

3.1.5.89 proc_SecGetUserPermissionOnGroup

The proc_SecGetUserPermissionOnGroup stored procedure is invoked to determine what permissions the requesting user has within a specified site group.

```sql
PROCEDURE proc_SecGetUserPermissionOnGroup(
    @SiteId uniqueidentifier,
    @GroupId int,
    @UserId int,
    @BelongsToGroup bit,
    @CanViewMembership bit OUTPUT,
    @CanEditMembership bit OUTPUT,
    @GroupOwnerId int OUTPUT,
    @IsExplicitlyInMembership bit OUTPUT,
    @RequestGuid uniqueidentifier OUTPUT
);
```

@SiteId: The site collection identifier of the site collection containing the site group specified by @GroupId.

@GroupId: The site group identifier of the site group whose permissions are to be returned.

@UserId: The user identifier of the current user making the request for permissions.

@BelongsToGroup: When this parameter is set to 1, it indicates that the user specified by @UserId belongs to the site group specified by @GroupId. This is used to determine access to site group information, as site groups can restrict membership information to only members of the site group.

@CanViewMembership: When this parameter is set to 1, it indicates that the user specified by @UserId is permitted to view site group membership.

@CanEditMembership: When this parameter is set to 1, it indicates that the user specified by @UserId is permitted to edit site group membership.

@GroupOwnerId: The user identifier of the user who owns the site group specified by @GroupId.

@IsExplicitlyInMembership: When this parameter is set to 1, it indicates that the user specified by @UserId is directly a member of the site group specified by @GroupId and not conferred membership through an intermediate site group membership.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetUserPermissionOnGroup stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>1319</td>
<td>The site group was not found for the provided @GroupId and @SiteId.</td>
</tr>
</tbody>
</table>

The proc_SecGetUserPermissionOnGroup stored procedure MUST return zero result sets.
proc_SecGetWebsAndListsWithUniquePermissions

The proc_SecGetWebsAndListsWithUniquePermissions stored procedure is invoked to return a list of sites and lists that have unique permissions.

PROCEDURE proc_SecGetWebsAndListsWithUniquePermissions(
@SiteId                   uniqueidentifier,
@WebId                    uniqueidentifier,
@WebUrl                   nvarchar(260),
@RequestGuid              uniqueidentifier OUTPUT
);

@SiteId: The site collection identifier of the site collection.
@WebId: The site identifier of the site (2) to query.
@WebUrl: The URL of the site collection.
@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecGetWebsAndListsWithUniquePermissions stored procedure returns an integer return code which MUST be 0.

Result Sets: proc_SecGetWebsAndListsWithUniquePermissions MUST return four result sets:

- Site Information Result Set (section 3.1.5.90.1)
- Sub Site Information Result Set (section 3.1.5.90.2)
- Site List Information Result Set (section 3.1.5.90.3)
- Sub Site List Information Result Set (section 3.1.5.90.4)

3.1.5.90.1 Site Information Result Set

The Site Information Result Set MUST return one row if the site has unique permissions. The Site Information Result Set MUST return 0 rows if the site (2) has no unique permissions.

    FullUrl nvarchar(256),
    Id     uniqueidentifier,
    Title  nvarchar(255);

FullUrl: The URL of the site (2) specified by @WebId.
Id: The site identifier of the site (2) specified by @WebId.
Title: The title associated with the site (2) specified by @WebId.

3.1.5.90.2 Sub Site Information Result Set

The Sub Site Information Result Set returns information about each subsite of the site specified by @WebId. The result set MUST return 0 or more rows. There MUST be one row per subsite that has unique permissions.

    FullUrl nvarchar(256),
    Id     uniqueidentifier,
    Title  nvarchar(255);
FullUrl: The URL of the subsite.

Id: The site identifier of the subsite.

Title: The title associated with the subsite.

3.1.5.90.3 Site List Information Result Set

The Site List Information Result Set returns information about lists in the site specified by @WebId. The number of rows returned MUST be the same as the number of lists which either have unique permissions or contain documents that have unique permissions.

```
FullUrl                      nvarchar(256),
Id                           uniqueidentifier,
Title                        nvarchar(255),
{ListRootFolderUrl}          nvarchar(260),
tp_ID                        uniqueidentifier,
tp_Title                     nvarchar(25),
HasUniquePerm                bit;
```

FullUrl: The URL of the site (2) specified by @WebId.

Id: The site identifier of the site (2) specified by @WebId.

Title: The title associated with the site (2) specified by @WebId.

{ListRootFolderUrl}: Root folder of the list which has unique permissions in the site (2) specified by @WebId.

tp_ID: The list identifier of the list.

tp_Title: The list title.

HasUniquePerm: This parameter is set to 0 if the list permissions is as the same as the site (2). This parameter is set to 1 if the list permissions is not as the same as the site (2).

3.1.5.90.4 Sub Site List Information Result Set

The Sub Site List Information Result Set returns information about lists in the subsite that is in the site (2) specified by the @WebId.

There MUST be a row for each list in the subsite of the site (2) specified by @WebId which either have unique permissions or contain documents that have unique permissions.

```
FullUrl                      nvarchar(256),
Id                           uniqueidentifier,
Title                        nvarchar(255),
{ListRootFolderUrl}          nvarchar(260),
tp_ID                        uniqueidentifier,
tp_Title                     nvarchar(25),
HasUniquePerm                bit;
```

FullUrl: The URL of the site (2) specified by the @WebId.

Id: The site identifier of the site (2) specified by the @WebId.
Title: The title associated with the site (2).

{ListRootFolderUrl}: Root folder of the list which has unique permissions in the site (2).

tp_ID: The list identifier of the list.

tp_Title: The list title.

HasUniquePerm: This parameter is set to 0 if the list permissions is as the same as the site (2). This parameter is set to 1 if the list permissions is not as the same as the site (2).

3.1.5.91  proc_SecListAllSiteMembers

The proc_SecListAllSiteMembers stored procedure provides information about all non-deleted security principals in the specified site collection.

PROCEDURE proc_SecListAllSiteMembers (  
    @SiteID                         uniqueidentifier,  
    @RequestGuid                    uniqueidentifier          OUTPUT  
);  

@SiteID: The site collection identifier of the site collection containing the requested security principal (2) information.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecListAllSiteMembers stored procedure returns an integer return code, which MUST be 0 and it MUST return a single result set.

3.1.5.91.1  User Information Result Set

The User Information Result Set returns information about the non-deleted users directly associated with the site collection specified by @SiteID. Zero or more rows MUST be returned. There MUST be one row per non-deleted user in the site collection specified by @SiteId.

tp_Id int,
.tp_SystemID varbinary(512),
.tp_Title nvarchar(255),
.tp_Login nvarchar(255),
.tp_Email nvarchar(255),
.tp_Notes nvarchar(1023),
.tp_SiteAdmin bit,
.tp_DomainGroup bit,
.tp_Flags int;

tp_Id: The user identifier that uniquely identifies the security principal with the site collection.

tp_SystemID: The SystemID of the security principal.

tp_Title: The display name of the security principal (2).

tp_Login: The login name of the security principal (2).

tp_Email: The email address of the security principal (2), for example, "username@mail.example.com".

tp_Notes: Notes associated with the security principal.
**tp_SiteAdmin**: This value is set to 1 if the security principal (2) is a site collection administrator. This value is set to 0 if the security principal (2) is not a site collection administrator.

**tp_DomainGroup**: This value is set to 1 if the security principal (2) is a domain group. This value is set to 0 if the security principal (2) is not a domain group.

**tp_Flags**: A 4-byte integer bit mask specifying the security principal's (2) options as specified in *UserInfo Flags* (section 2.2.2.12).

### 3.1.5.92  proc_SecListAllWebMembers

The **proc_SecListAllWebMembers** stored procedure is invoked to list all non-deleted user and domain group accounts registered with a specified **site (2)**.

```sql
PROCEDURE proc_SecListAllWebMembers(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @RequestGuid                   uniqueidentifier          OUTPUT
);
```

- **@SiteId**: The site collection identifier of the site collection containing the site (2) specified by **@WebId**.
- **@WebId**: The site identifier of the site (2) to query for **members**.
- **@RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_SecListAllWebMembers** stored procedure returns an integer return code which MUST be 0 and it MUST return a single result set.

#### 3.1.5.92.1 Site Membership Result Set

The **Site Membership Result Set** returns a list of all non-deleted users or domain groups registered with the specified **site (2)**. The **Site Membership Result Set** MUST return zero rows if the site specified by **@WebId** has no **members**, otherwise it MUST contain one row for each member if **@WebId** matches an existing site (2) within the site collection specified by **@SiteId**.

```sql
tp_ID                          int,
tp_SystemID                    varbinary(512),
tp_Title                       nvarchar(255),
tp_Login                       nvarchar(255),
tp_Email                       nvarchar(255),
tp_Notes                       nvarchar(1023),
tp_SiteAdmin                   bit,
tp_DomainGroup                 bit,
tp_Flags                       int;
```

- **tp_ID**: The user identifier of the **security principal**.
- **tp_SystemID**: The **SystemID** of the security principal.
- **tp_Title**: The display name of the security principal (2).
- **tp_Login**: The login name of the security principal (2).
- **tp_Email**: The email address of the security principal.
- **tp_Notes**: A descriptive text associated with the security principal (2).
tp_SiteAdmin: Contains a bit flag set to 1 if the security principal (2) is a site collection administrator. This value is set to 0 if the user is not a site collection administrator.

tp_DomainGroup: Contains a bit flag set to 1 if the security principal (2) is a domain group. This value is set to 0 if the security principal (2) is not for a domain group.

tp_Flags: A 4-byte integer bit mask determining the security principal's (2) options as specified in UserInfo Flags (section 2.2.2.12).

3.1.5.93  proc_SecListGroupsInRole

The proc_SecListGroupsInRole stored procedure is invoked to list all site groups that have the specified role.

PROCEDURE proc_SecListGroupsInRole(
    @SiteId                        uniqueidentifier,
    @ScopeId                       uniqueidentifier,
    @RoleId                        int,
    @RequestGuid                   uniqueidentifier
);

@SiteId: The Site Collection Identifier of the site collection that contains the site groups that have the role.

@ScopeId: The Scope Identifier of the scope containing the role.

@RoleId: Specifies the Role Identifier for the role.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecListGroupsInRole stored procedure returns an integer return code which MUST be 0.

The proc_SecListGroupsInRole stored procedure MUST return 1 result set.

3.1.5.93.1  Site Group Information Result Set

The Site Group Information Result Set returns the collection of site groups that have the specified role. The Site Group Information Result Set MUST be empty if either the site collection, role, or scope are not found, or if no site groups have the specified role in the specified scope. Otherwise, the Site Group Information Result Set MUST have one row per site group. The Site Group Information Result Set schema is defined in the Sec_SiteGroupsView (section 2.2.6.5).

3.1.5.94  proc_SecListScopeGroups

The proc_SecListScopeGroups stored procedure is invoked to list all site groups that have role assignments in a specified scope.

PROCEDURE proc_SecListScopeGroups(
    @SiteId                        uniqueidentifier,
    @ScopeId                       uniqueidentifier,
    @RequestGuid                   uniqueidentifier
);

@SiteId: The Site Collection Identifier of the site collection which contains the scope whose site groups with role assignments are being listed.

@ScopeId: The Scope Identifier of the specified scope.
@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecListScopeGroups stored procedure returns an integer return code which MUST be 0.

The proc_SecListScopeGroups stored procedure MUST return a single result set as defined in the following section.

3.1.5.94.1 Site Groups Result Set

The Site Groups Result Set contains all site group information for site groups with a role assignment in the specified scope. The Site Groups Result Set MUST be empty if either the site collection or scope are not found, or if no site groups have role assignments in the specified scope. Otherwise, one row MUST be returned for each site group in the site collection which has a role assignment in the specified scope.

The Site Groups Result Set is the same as the view defined in the Sec_SiteGroupsView (section 2.2.6.5).

3.1.5.95 proc_SecListScopeUsers

The proc_SecListScopeUsers stored procedure is invoked to list information about users and domain groups with a direct role assignment association with the specified scope, rather than by membership in a site group with a role assignment association with the scope.

PROCEDURE proc_SecListScopeUsers(
    @SiteId uniqueidentifier,
    @ScopeId uniqueidentifier,
    @RequestGuid uniqueidentifier
);

@SiteId: The site collection identifier of the site collection containing the scope.

@ScopeId: The scope identifier of the scope for which information about users with role assignments is requested.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecListScopeUsers stored procedure MUST return an integer return code of 0, and it MUST return a single result set with 0 or more rows.

3.1.5.95.1 User Information Result Set

The User Information Result Set returns information about the users and domain groups directly associated with the specified scope through a role assignment. The User Information Result Set MUST contain 1 row for each non-deleted user or domain group directly associated with the specified scope through a role assignment. The User Information Result Set MUST contain 0 rows if no users or domain groups are directly associated with the specified scope, or if the parameters @SiteId or @ScopeId do not match existing site collection or scope identifiers, respectively.

The User Information Result Set is defined in the Common Result Sets Principal User Information Result Set (section 2.2.4.19).

3.1.5.96 proc_SecListSiteGroupMembership

The proc_SecListSiteGroupMembership stored procedure lists members in a specified site group. To view membership, the site group MUST be set to allow everyone to view membership, or the current user MUST belong to the site group, be a site auditor, or be the site owner.
PROCEDURE proc_SecListSiteGroupMembership (  
  @SiteId uniqueidentifier,  
  @GroupId int,  
  @CurrentUserId int,  
  @SiteAuditor bit,  
  @BelongsToGroup bit,  
  @GroupOwnerId int,  
  @CurrentUserIsOwner bit,  
  @RequestGuid uniqueidentifier  
);  

@SiteId: The site collection identifier of the site collection containing the site group.  

@GroupId: The identifier of the site group.  

@CurrentUserId: The identifier of the current user. This is used to determine privileges.  

@SiteAuditor: A bit set to 1 if the current user is a site auditor. This is used to determine privileges.  

@BelongsToGroup: A bit set to 1 if the current user is a member of the site group. This is used to determine privileges.  

@GroupOwnerId: The identifier of the owner of the site group. This is used to determine privileges.  

@CurrentUserIsOwner: A bit set to 1 if the current user is the owner of the site group. This is used to determine privileges.  

@RequestGuid: The optional request identifier for the current request.  

Return Values: The proc_SecListSiteGroupMembership stored procedure returns an integer return code which MUST be listed in the following table.  

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>5</td>
<td>The User does not have privileges to view site group membership.</td>
</tr>
</tbody>
</table>

The proc_SecListSiteGroupMembership stored procedure MUST return a single result set.  

3.1.5.96.1 User Information Result Set  

The User Information Result Set MUST either be empty or include 1 row of information for each member of the site group. If the user specified by @CurrentUserId does not have privileges to view membership, then the User Information Result Set will be empty. To view membership, the site group MUST be set to allow everyone to view membership, or the current user MUST belong to the site group, be a site auditor, or be the site owner.  

The User Information Result Set is defined the Common Result Sets Principal User Information Result Set (section 2.2.4.19).  

3.1.5.97 proc_SecListSiteGroups  

The proc_SecListSiteGroups stored procedure is invoked to list site group information for a specified site collection.  

PROCEDURE proc_SecListSiteGroups(  
  @SiteId uniqueidentifier,  
  @RequestGuid uniqueidentifier  
);
@SiteId: The site collection identifier of the site collection containing the site groups to list.

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_SecListSiteGroups` stored procedure returns an integer return code, which MUST be 0, and MUST return one result set.

### 3.1.5.97.1 Site Group Information Result Set

The Site Group Information Result Set contains all site group information for the site collection. The Site Group Information Result Set MUST be empty if the site collection is not found. Otherwise, 1 row MUST be returned for each site group in the specified site collection. The Site Group Information Result Set is defined using T-SQL syntax, in the Sec_SiteGroupsView (section 2.2.6.5).

### 3.1.5.98 proc_SecListSiteGroupsContainingUser

The `proc_SecListSiteGroupsContainingUser` stored procedure lists the site groups in which the user is a member.

```
PROCEDURE proc_SecListSiteGroupsContainingUser(
    @SiteId uniqueidentifier,
    @UserId int,
    @RequestGuid uniqueidentifier
);
```

@SiteId: The site collection identifier of the site collection containing the site groups to be listed. This parameter MUST correspond to a valid site collection, and MUST NOT be NULL.

@UserId: The user identifier of the current user. This is used to determine site groups containing this user.

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_SecListSiteGroupsContainingUser` stored procedure returns an integer return code which MUST be 0. The `proc_SecListSiteGroupsContainingUser` stored procedure MUST return one result set.

### 3.1.5.98.1 Site Group Information Result Set

The Site Group Information Result Set contains information on site groups where the user is a member. The Site Group Information Result Set MUST be returned, and MUST contain 1 row for each site group found for the user. The Site Group Information Result Set schema is defined in the Sec_SiteGroupsView (section 2.2.6.5).

### 3.1.5.99 proc_SecListSiteGroupsWhichUserOwns

The `proc_SecListSiteGroupsWhichUserOwns` stored procedure returns site group information for the site groups owned by the specified principal (1).

```
PROCEDURE proc_SecListSiteGroupsWhichUserOwns(
    @SiteId uniqueidentifier,
    @UserId int,
    @RequestGuid uniqueidentifier
);
```
@SiteId: The site collection identifier of the site collection containing the principal and the site groups to be listed.

@UserId: The user identifier of the principal, used to find the site groups it owns.

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_SecListSiteGroupsWhichUserOwns` stored procedure returns an integer return code which MUST be 0, and MUST return one result set.

### 3.1.5.99.1 Site Group Information Result Set

The **Site Group Information Result Set** returns site group information for all site groups which are owned either directly by the user or external group specified by @UserId, or for site groups which are owned indirectly, when the user or external group is a member of a site group which is the returned site group's owner.

The **Site Group Information Result Set** MUST contain one row for each site group owned. The T-SQL syntax for the **Site Group Information Result Set** is defined in the Sec_SiteGroupsView (section 2.2.6.5).

### 3.1.5.100 proc_SecListUsersInRole

The **proc_SecListUsersInRole** stored procedure lists the non-deleted principals assigned to a specified role in the specified scope.

```
PROCEDURE proc_SecListUsersInRole(
    @SiteId                        uniqueidentifier,
    @ScopeId                       uniqueidentifier,
    @RoleId                        int,
    @RequestGuid                   uniqueidentifier
);
```

@SiteId: The site collection identifier of the site collection containing the scope, role, and principals to be listed.

@ScopeId: The scope identifier of the scope within which to match principals assigned to the role.

@RoleId: Specifies the role identifier of the role definition assigned to the principals to be listed.

@RequestGuid: The optional request identifier for the current request.

Return Values: The **proc_SecListUsersInRole** stored procedure returns an integer return code which MUST be 0, and MUST return one result set.

### 3.1.5.100.1 User Role Membership Result Set

The **User Role Membership Result Set** returns information about the non-deleted principals assigned to the specified role in the specified scope and site collection. The **User Role Membership Result Set** MUST be empty if no principals assigned to the role are found; otherwise, it MUST include one row of information for each matching principal (1).

```
tp_Id                          int,
tp_SystemID                    varbinary(512),
tp_Title                       nvarchar(255),
```
tp_Id: The user identifier of the principal.

tp_SystemID: The SystemID of the principal.

tp_Title: The display name of the principal.

tp_Login: The login name of the principal.

tp_Email: The email address of the principal.

tp_Notes: Contains notes about the principal.

tp_SiteAdmin: If the parameter is set to 1, then the principal is a site collection administrator. If the parameter is set to 0, then the principal is not a site collection administrator.

tp_DomainGroup: If the parameter is set to 1, the principal is a domain group. If the parameter is set to 0, the principal is not a domain group.

tp_Flags: The user information flags value of the principal.

3.1.5.101 proc_SecMigrateUser

The proc_SecMigrateUser stored procedure updates the SystemID and login name for a principal (1) in the UserInfo (section 2.2.6.10) and AllUserData (section 2.2.6.2) Tables. If an existing principal is found with the new login name and SystemID, then proc_SecMigrateUser MUST update that principal with a new unique SystemID and mark that principal as deleted in the UserInfo table before updating the specified principal. If an existing principal is found with the new login name and SystemId, then proc_SecMigrateUser MUST also mark that principal as deleted in the AllUserData table's user list.

PROCEDURE proc_SecMigrateUser(
    @SiteId uniqueidentifier,
    @OldLogin nvarchar(255),
    @NewLogin nvarchar(255),
    @NewSystemId varbinary(512),
    @IsWindowsAuth bit,
    @PeopleListId uniqueidentifier,
    @LoginColumnName nvarchar(64),
    @DeletedColumnName nvarchar(64),
    @OldSystemId varbinary(512) = NULL,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);

@SiteId: The site collection identifier of the site collection containing the principal to be updated.

@OldLogin: The current value of the principal's login name.

@NewLogin: The new value to set as the principal's login name. This parameter MUST NOT be NULL.

@NewSystemId: The new SystemID to set for the principal. This parameter MUST NOT be NULL.
@IsWindowsAuth: A bit specifying whether the principal is authenticated with Windows-integrated authentication. If this parameter is set to 1 or NULL, then the principal is authenticated using Windows-integrated authentication.

@PeopleListId: The list identifier of the user list.

@LoginColumnName: The name of the column in the AllUserData table that stores the user list's login names.

@DeletedColumnName: The name of the column in the AllUserData table that marks whether a principal in the user list has been deleted.

@OldSystemId: The current value of the principal's SystemID. This parameter can be NULL. If the @OldSystemId parameter is not NULL, then the principal specified by @OldLogin MUST also have the SystemID specified by @OldSystemId.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecMigrateUser stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>1003</td>
<td>An existing principal was found with the new login name and SystemID, but proc_SecMigrateUser failed to update it with a new SystemID and mark it as deleted. Failed to find or update the specified principal.</td>
</tr>
</tbody>
</table>

The proc_SecMigrateUser stored procedure MUST return no result sets.

3.1.5.102 proc_SecReCalculateWebFGP

The proc_SecReCalculateWebFGP stored procedure is invoked to update the bit at 0x00000400 of the Site Property Flags (section 2.2.2.11) to indicate whether the site (2) has at least one uniquely secured object within it. This bit can be used by a search tool to determine whether it avoids indexing pages in this site (2) so as not to reveal secure information.

PROCEDURE proc_SecReCalculateWebFGP(
    @SiteId uniqueidentifier,
    @WebId uniqueidentifier,
    @RequestGuid uniqueidentifier
);

@SiteId: The site collection identifier of the site collection containing the site (2) to be updated. This parameter MUST correspond to a valid site collection, and MUST NOT be NULL.

@WebId: The site identifier of the site (2) containing the flag to be updated.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecReCalculateWebFGP stored procedure returns an integer return code, which MUST be 0, and it MUST return zero result sets.

3.1.5.103 proc_SecRefreshToken

The proc_SecRefreshToken stored procedure is invoked to update the UserInfo Table (section 2.2.6.10) with information about a specified user's membership in external groups.
PROCEDURE proc_SecRefreshToken(
    @SiteId                        uniqueidentifier,
    @UserId                        int,
    @ExternalToken                 varbinary(max),
    @ExternalTokenTime             datetime,
    @RequestGuid                   uniqueidentifier
);

@SiteId: The site collection identifier of the site collection containing the user to update in the UserInfo Table (section 2.2.6.10).

@UserId: The user identifier of the user.

@ExternalToken: An External Group Token (section 2.2.3.2) containing external group membership information for the user. This value can be NULL, specifying that the user is not a member of any external groups. proc_SecRefreshToken MUST update the UserInfo Table row matching the user identifier with this value in the tp_ExternalToken field if the table row value in tp_ExternalTokenLastUpdated is not greater than the value of @ExternalTokenTime.

@ExternalTokenTime: A datetime in UTC specifying when the External Group Token in the @ExternalToken parameter was last updated. This parameter MUST be NULL if the @ExternalToken parameter is NULL. proc_SecRefreshToken MUST update the UserInfo Table row matching the user identifier with this value in the tp_ExternalTokenLastUpdated field if the table row value in tp_ExternalTokenLastUpdated is not greater than the value of @ExternalTokenTime.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecRefreshToken stored procedure returns an integer return code, which MUST be 0, and it MUST NOT return a result set.

3.1.5.104 proc_SecRemoveGroup

The proc_SecRemoveGroup stored procedure is invoked to remove a site group from a site collection.

PROCEDURE proc_SecRemoveGroup(
    @SiteId                        uniqueidentifier,
    @GroupId                       int,
    @UserID                        int,
    @AppPrincipalId                int,
    @SiteAdmin                     bit,
    @GroupOwnerId                  int,
    @CurrentUserIsOwner            bit,
    @ReturnDLAlias                 bit,
    @CheckIfInUse                  bit,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);

@SiteId: The site collection identifier of the site collection containing the site group to be removed.

@GroupId: The site group identifier for the site group to be removed.

@UserID: The user identifier for the current user. The ownership of any site group previously owned by the deleted site group MUST be re-assigned to this user identifier.

@AppPrincipalId: The app principal identifier of the app that is requesting this operation. If the operation was not requested by an app then it MUST be set to 0.
@SiteAdmin: A bit flag specifying whether the current user is a site collection administrator. If this flag is set to 1, then the site group MUST be removed. If this flag is set to 0, then the site group MUST NOT be removed unless @CurrentUserIsOwner specifies it.

@GroupOwnerId: The User Identifier or Site Group Identifier of the site group owner. If @CurrentUserIsOwner is set to 1, then this value MUST be set to the User Identifier of the current user, or a site group which includes the current user. If @CurrentUserIsOwner is set to 0, then this value MUST be ignored.

@CurrentUserIsOwner: A bit flag specifying that the current user is an owner of the site group. If @CurrentUserIsOwner is set to 1 and @GroupOwnerId contains the User Identifier of the owner of the site group, or the Site Group Identifier of a site group, which includes the current user, then the site group MUST be removed. If @CurrentUserIsOwner is set to 0, then the site group MUST NOT be removed unless @SiteAdmin specifies it.

@ReturnDLAlias: A bit flag specifying whether to return the DLAlias Result Set. If this flag is set to 1, the result set MUST be returned. If this flag is set to 0, then the result MUST NOT be returned.

@CheckIfInUse: A bit flag specifying whether to check if the site group is in use before removing it. A site group is in use if it is assigned to any role in the site collection. If the site group is not in use, it is removed. If the site group is in use, then it MUST NOT be removed, but the stored procedure MUST return an empty DLAlias Result Set if @ReturnDLAlias is 1 and a Return Value of 0.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecRemoveGroup stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>5</td>
<td>The current user has insufficient permissions to remove the site group.</td>
</tr>
</tbody>
</table>

The proc_SecRemoveGroup stored procedure returns either zero or one result set. If the site group is successfully removed and @ReturnDLAlias is set to 1, then proc_SecRemoveGroup MUST return a result set. If the current user does not have sufficient permissions to remove the site group or @ReturnDLAlias is set to 0, then proc_SecRemoveGroup MUST NOT return a result set.

### 3.1.5.104.1 DLAlias Result Set

If the site group existed and was removed successfully and if @ReturnDLAlias is set to 1, the DLAlias Result Set MUST return the email distribution address of the site group in a single row. The DLAlias Result Set MUST be empty if the email distribution address was blank.

```sql
DLAlias              varchar(128);
```

DLAlias: The e-mail distribution address for the site group.

### 3.1.5.105 proc_SecMarkGroupForWebDelete

The proc_SecMarkGroupForWebDelete stored procedure is invoked to mark a site group for later removal from a site collection.

```sql
PROCEDURE proc_SecMarkGroupForWebDelete(
    @SiteId uniqueidentifier,
    @GroupId int,
)
```
@SiteId: The site collection identifier (section 2.2.1.9) of the site collection containing the site group to be marked for removal.

@GroupId: The site group identifier for the site group to be marked for removal.

@UserID: The user identifier for the current user.

@AppPrincipalId: The app principal identifier of the app requesting this operation. If the request is not made by an app then it MUST be set to 0.

@SiteAdmin: A bit flag specifying whether the current user is a site collection administrator. If this flag is set to 1, then the site group MUST be marked for removal. If this flag is set to 0, then the site group MUST NOT be marked for removal unless @CurrentUserIsOwner specifies it.

@GroupOwnerId: The user identifier or site group identifier of the site group owner. If @CurrentUserIsOwner is set to 1, then this value MUST be set to the user identifier of the current user, or a site group that includes the current user. If @CurrentUserIsOwner is set to 0, then this value MUST be ignored.

@CurrentUserIsOwner: A bit flag specifying that the current user is an owner of the site group. If @CurrentUserIsOwner is set to 1 and @GroupOwnerId contains the user identifier of the owner of the site group, or the site group identifier of a site group that includes the current user, then the site group MUST be marked for removal. If @CurrentUserIsOwner is set to 0, then the site group MUST NOT be marked for removal unless @SiteAdmin specifies it.

@DeletionWebId: The site identifier of the site (2) associated with the group.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecMarkGroupForWebDelete stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>5</td>
<td>The current user has insufficient permissions to mark the site group for removal.</td>
</tr>
</tbody>
</table>

The proc_SecMarkGroupForWebDelete stored procedure returns either zero or one result set. If the site group is successfully marked for removal, then proc_SecMarkGroupForWebDelete MUST return a result set. If the current user does not have sufficient permissions to mark the site group for removal, then proc_SecMarkGroupForWebDelete MUST NOT return a result set.

3.1.5.105.1 DLAlias Result Set

If the site group existed and was marked for removal successfully, the DLAliases Result Set MUST return the e-mail distribution address of the site group in a single row. The DLAliases Result Set MUST be empty if the e-mail distribution address was blank.
**DLAlias**: The e-mail distribution address for the site group.

### 3.1.5.106 proc_SecRemovePrincipalFromScope

The **proc_SecRemovePrincipalFromScope** stored procedure removes a principal from a security role associated with a site (2) in a specified security scope. If no role is specified, or if the role specified is the "Guest" Role, then the principal MUST be removed from all roles in the scope, and it MUST also be removed from all roles in other specified scopes or subsite contained within the Site.

```
PROCEDURE proc_SecRemovePrincipalFromScope (  
    @SiteId                        uniqueidentifier,  
    @WebId                         uniqueidentifier,  
    @WebScopeId                    uniqueidentifier,  
    @ScopeId                       uniqueidentifier,  
    @RemoveFromCurrentScopeOnly    bit,  
    @PrincipalId                   int,  
    @RoleId                        int = NULL,  
    @RequestGuid                   uniqueidentifier = NULL OUTPUT  
);
```

- **@SiteId**: The Site Collection Identifier of the site collection containing the principal and scope.
- **@WebId**: The Site Identifier of the site (2) containing the principal and scope.
- **@WebScopeId**: The Scope Identifier of the scope which includes the Site. If the Site has unique permissions, then this MUST be the scope of the Site. If **@RoleId** is not NULL and not the Role Identifier of the Guest Role (1073741825), then **@WebScopeId** MUST be ignored. Otherwise, **@WebScopeId** determines additional scopes or Sites from which the principal MUST be removed, as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@WebScopeId = @ScopeId</td>
<td>Remove the principal from the specified role or roles within the Site and within all of the Site's subsites which share the same scope.</td>
</tr>
<tr>
<td>@WebScopeId != @ScopeId</td>
<td>Remove the principal from the specified role or roles in all scopes contained within the Site.</td>
</tr>
</tbody>
</table>

- **@ScopeId**: The Scope Identifier of the scope in which to remove the principal from the role. See **@WebScopeId** for behavior based on comparison of the **@ScopeId** and **@WebScopeId** parameter values.

- **@RemoveFromCurrentScopeOnly**: The bit flag indicating from which scope the principal will be removed. If this flag is set to 1, then **proc_SecRemovePrincipalFromScope** MUST remove the principal from the current scope only.

- **@PrincipalId**: The User Identifier of the principal to remove.

- **@RoleId**: The Role Identifier of the role to remove the principal from, within the specified scopes. The value in **@RoleId** MUST either correspond to a valid role or be NULL. If this value is NULL, or if **@RoleId** is the Role Identifier of the Guest Role (1073741825), then **proc_SecRemovePrincipalFromScope** MUST remove the principal from all roles in the specified scopes.

- **@RequestGuid**: The optional request identifier for the current request.
Return Values: The `proc_SecRemovePrincipalFromScope` stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The Scope Identifier points to a Scope whose path cannot be found.</td>
</tr>
</tbody>
</table>

The `proc_SecRemovePrincipalFromScope` stored procedure MUST return a single result set as follows.

### 3.1.5.106.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains the information about the Audit Flags (section 2.2.2.1) associated with the specified Site. The Site Audit Mask Result Set MUST be returned and MUST contain a single row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).

### 3.1.5.107 proc_SecRemoveRoleDef

The `proc_SecRemoveRoleDef` stored procedure removes a role definition, and any role assignments which use that role definition, from a specified site (2).

```sql
PROCEDURE proc_SecRemoveRoleDef(
    @SiteId    uniqueidentifier,
    @WebId     uniqueidentifier,
    @RoleId    int,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection containing the role definition to be removed.

@WebId: The site identifier of the site associated with the role definition to be removed.

(RoleId): The identifier of the role definition to be removed.

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_SecRemoveRoleDef` stored procedure MUST return an integer return code of 0. The `proc_SecRemoveRoleDef` stored procedure MUST return the following result set.

### 3.1.5.107.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the Site. The Site Audit Mask Result Set MUST be returned and MUST contain one row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).

### 3.1.5.108 proc_SecRemoveUserFromScopeByLogin

The `proc_SecRemoveUserFromScopeByLogin` stored procedure is invoked to remove a user, specified by login name, from a role in a specified security scope. If no role is specified, or if the role specified is the "Guest" Role, then the user will be removed from all roles in the specified scope and will also be removed from other scopes associated with the specified site (2).

```sql
PROCEDURE proc_SecRemoveUserFromScopeByLogin(

```
@SiteId: The site collection identifier of the site collection containing the user and scope.

@WebId: The site identifier of the site (2) using the role.

@WebScopeId: The Scope Identifier of the scope which includes the site (2) specified by @WebId. If @RoleId is not NULL and is not the role identifier of the "Guest" role (1073741825), then @WebScopeId MUST be ignored. Otherwise, @WebScopeId determines which additional scopes the User MUST be removed from, as follows:

- If @WebScopeId is equal to @ScopeId, remove the specified User from all scopes associated with all sites sharing the permissions of the site (2) specified by @WebId.
- If @WebScopeId is not equal to @ScopeId, remove the specified user from all scopes associated with the site (2) specified by @WebId.

@ScopeId: The scope identifier of the scope in the site collection from which to remove the user. See @WebScopeId for conditions of equality between @ScopeId and @WebScopeId.

@RemoveFromCurrentScopeOnly: The bit flag indicating from which scope the user is removed. If this flag is set to 1, then proc_SecRemoveUserFromScopeByLogin MUST remove the user from the current scope only.

@LoginName: The login name of the user.

@RoleId: The role identifier to remove from the specified scopes. @RoleId MUST be NULL or correspond to a valid role. If @RoleId is NULL, or is the role identifier of the Guest role (1073741825), then proc_SecRemoveUserFromScopeByLogin MUST remove the User corresponding to @LoginName from all roles.

@RequestId: The optional request identifier for the current request.

Return Values: The proc_SecRemoveUserFromScopeByLogin stored procedure returns an integer return code, which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>1317</td>
<td>UserId not found for site collection or login name.</td>
</tr>
</tbody>
</table>

The proc_SecRemoveUserFromScopeByLogin stored procedure MUST return a single result set as follows.

### 3.1.5.108.1 Site Audit Mask Result Set

The Site Audit Mask Result Set returns audit information about the Site specified by @WebId. The Site Audit Mask Result Set MUST be returned, and MUST contain only one row, as defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).
The `proc_SecRemoveUserFromSite` stored procedure is invoked to remove a user from a site collection. If the User owns site groups, then `proc_SecRemoveUserFromSite` will assign site group ownership to the specified new group owner.

```sql
PROCEDURE proc_SecRemoveUserFromSite(
    @SiteId uniqueidentifier,  
    @NewSiteGroupOwnerId int,  
    @UserId int,  
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The Site Collection Identifier for the site collection containing the user to be removed.

@NewSiteGroupOwnerId: An identifier for the user who will take over site group ownership from the user being removed. To succeed, this value MUST be -1 or a valid User Identifier in the specified site collection, and @NewSiteGroupOwnerId MUST NOT have the same value as @UserId. If the value is -1, or system account (1073741823) or app account (1073741822) then the site collection owner will be the new site group owner.

@UserId: The User Identifier for the user who is being removed. If @UserId is the site collection owner or secondary contact, the change will not succeed. To succeed, this value MUST be an existing User Identifier for the specified site collection.

@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_SecRemoveUserFromSite` stored procedure returns an integer return code which MUST be included in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 is returned whenever the conditions for the alternate return values of 16 and 4335 are not met.</td>
</tr>
<tr>
<td>16</td>
<td>@NewSiteGroupOwnerId equals @UserId. No changes were made.</td>
</tr>
<tr>
<td>4335</td>
<td>@UserId is the site collection owner or secondary contact. No changes were made.</td>
</tr>
</tbody>
</table>

The `proc_SecRemoveUserFromSite` stored procedure MUST NOT return any result sets.

The `proc_SecRemoveUserFromSiteGroup` stored procedure is invoked to remove a user from a site group in a site collection.

```sql
PROCEDURE proc_SecRemoveUserFromSiteGroup(
    @SiteId uniqueidentifier,  
    @GroupId int,  
    @UserIDToDelete int,  
    @UserId int,  
    @AppPrincipalId int,  
    @SiteAdmin bit,  
    @BelongsToGroup bit,  
    @GroupOwnerId int,  
    @CurrentUserIsOwner bit,  
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```
@SiteId: The **site collection identifier** (section 2.2.1.1.9) of the site collection that will be queried for the requested site group.

@GroupId: The site group identifier for the site group that the user specified by @UserIDToBeDeleted is to be removed from.

@UserIDToBeDeleted: The user identifier for the user to remove from the specified site group. If @UserId and @UserIDToBeDeleted refer to the same user, and the site group permits users to remove themselves from site group membership, then the user specified by @UserIDToBeDeleted MUST be removed from the group (2).

@UserID: The user identifier for the current user who is removing the user to be deleted from the specified site group. If @UserId and @UserIDToBeDeleted refer to the same user, and the site group permits site group **members** to edit membership, then the user specified by @UserIDToBeDeleted is removed from the site group.

@AppPrincipalId: The **app principal** identifier of the **app** that is requesting this remove operation. If the request is not made by an app then it MUST be set to 0.

@SiteAdmin: A bit flag specifying whether the user specified by @UserID is a site collection administrator of the site collection specified by @SiteId. If @SiteAdmin is set to "1", then proc_SecRemoveUserFromSiteGroup MUST remove the user specified by @UserIDToBeDeleted from the site group.

@BelongsToGroup: A bit flag specifying whether the user specified by @UserID is a member of the site group specified by @GroupId. If @BelongsToGroup is set to "1" and the site group permits site group members to edit membership, and the current user specified in @UserID is a member of the site group, then proc_SecRemoveUserFromSiteGroup MUST remove the user specified by @UserIDToBeDeleted from the site group.

@GroupOwnerId: The user identifier or site group identifier of the site group owner specified by @GroupId. @GroupOwnerId MUST be set to the user identifier of the current user, or, if @CurrentUserIsOwner is set to "1," to the site group identifier of a site group which includes the current user. If @CurrentUserIsOwner is set to "0", then @GroupOwnerId MUST be ignored.

@CurrentUserIsOwner: A bit flag specifying that the user specified by @GroupOwnerId is an owner of the site group specified by @GroupId. If @CurrentUserIsOwner is set to "1", and @GroupOwnerId contains either the user identifier of the owner of the site group, or the site group identifier of a site group which includes the current user, then proc_SecRemoveUserFromSiteGroup MUST remove the user from the site group.

@RequestGuid: The optional request identifier for the current request.

**Return Values:** The proc_SecRemoveUserFromSiteGroup stored procedure returns an integer return code, which MUST be one of the values in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>5</td>
<td>The current user has insufficient authority to remove the specified user from the site group.</td>
</tr>
</tbody>
</table>

The proc_SecRemoveUserFromSiteGroup stored procedure MUST NOT return any result sets.

### 3.1.5.111 proc_SecRemoveUserFromSiteGroupByLogin

The proc_SecRemoveUserFromSiteGroupByLogin stored procedure is invoked to remove a user identified by login name from a specified site group.
PROCEDURE proc_SecRemoveUserFromSiteGroupByLogin(
    @SiteId                        uniqueidentifier,
    @GroupId                       int,
    @LoginName                     nvarchar(255),
    @UserID                        int,
    @SiteAdmin                     bit,
    @BelongsToGroup                bit,
    @GroupOwnerId                  int,
    @CurrentUserIsOwner            bit,
    @RequestGuid                   uniqueidentifier   = NULL OUTPUT
);

@SiteId: The Site Collection Identifier of the site collection containing the site group and the User to be removed.

@GroupId: The identifier of the site group containing the User to be removed.

@LoginName: The login name of the User to be removed from the site group.

@UserID: The User Identifier of the current user.

@SiteAdmin: A bit flag specifying whether the User specified by @UserID is a site collection administrator of the site collection specified by @SiteId. If @SiteAdmin is set to "1", and @LoginName specifies an existing user, then proc_SecRemoveUserFromSiteGroupByLogin MUST remove the User specified by @LoginName from the site group.

@BelongsToGroup: A bit flag specifying whether the User specified by @UserID is a member of the site group specified by @GroupId. If @BelongsToGroup is set to "1" and the site group permits users to manage site group memberships, and the current user specified in @UserID is a member of the site group, and @LoginName specifies an existing user, then proc_SecRemoveUserFromSiteGroupByLogin MUST remove the User specified by @LoginName from the site group.

@GroupOwnerId: The User Identifier or Site Group Identifier of the site group owner specified by @GroupId. @GroupOwnerId MUST be set to the User Identifier of the current user, or, if @CurrentUserIsOwner is set to "1," to the Site Group Identifier of a site group which includes the current user. If @CurrentUserIsOwner is set to "0", @GroupOwnerId MUST be ignored.

@CurrentUserIsOwner: A bit flag specifying that the User specified by @GroupOwnerId is an owner of the site group specified by @GroupId. If @CurrentUserIsOwner is set to "1" and @GroupOwnerId contains either the User Identifier of the owner of the site group, or the site group identifier of a site group which includes the current user, and @LoginName specifies an existing user, then proc_SecRemoveUserFromSiteGroupByLogin MUST remove the User specified by @LoginName from the site group.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecRemoveUserFromSiteGroupByLogin stored procedure returns an integer return code, which MUST be included in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>5</td>
<td>The current user has insufficient permissions to remove the specified User from the site group.</td>
</tr>
<tr>
<td>1317</td>
<td>UserId not found for site collection or login name.</td>
</tr>
</tbody>
</table>

The proc_SecRemoveUserFromSiteGroupByLogin stored procedure MUST NOT return a result set.
3.1.5.112 proc_SecResetItemPerm

The **proc_SecResetItemPerm** stored procedure is invoked to cause a list item to revert its set of permissions so that it inherits permissions from its parent rather than has its own. The previous set of unique permissions for the list item are discarded.

```sql
PROCEDURE proc_SecResetItemPerm(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @OldScopeId                    uniqueidentifier,
    @Url                           nvarchar(260),
    @DocId                         uniqueidentifier,
    @NewScopeId                    uniqueidentifier = NULL OUTPUT,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The **Site Collection Identifier** (section 2.2.1.1.9) of the **site collection** containing the list item.

@WebId: The **Site Identifier** (section 2.2.1.1.11) of the **site (2)** containing the item.

@OldScopeId: The **Scope Identifier** (section 2.2.1.1.8) of the **security scope** of the item.

@Url: The **store-relative form URL** of the item.

@DocId: The **document identifier** of the document.

@NewScopeId: This is an output parameter, which provides the **Scope Identifier** of the **security scope** of the item. This MUST be the **Scope Identifier** of the item's parent container upon successful execution; otherwise, it remains unchanged.

@RequestGuid: The optional **request identifier** for the current request.

Return Values: The **proc_SecResetItemPerm** stored procedure returns an integer return code which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The item specified is not configured to have unique permissions. This value MUST also be returned when the list item is not found at the specified location, or the @OldScopeID does not match the specified list item's security scope.</td>
</tr>
</tbody>
</table>

Upon successful execution, the **proc_SecResetItemPerm** stored procedure MUST return the **Site Audit Mask Result Set** (section 3.1.5.112.1).

3.1.5.112.1 Site Audit Mask Result Set

The **Site Audit Mask Result Set** contains the information about the **Audit Flags** (section 2.2.2.1) associated with the specified Site. It MUST contain a single row. The **Site Audit Mask Result Set** is defined using T-SQL syntax, as defined in the Common Result Sets **Site Audit Mask Result Set** (section 2.2.4.22).

3.1.5.113 proc_SecResetWebToDefaultRoleDefinition

The **proc_SecResetWebToDefaultRoleDefinition** stored procedure is invoked to replace existing roles, role assignments, and Permissions for a specified **site (2)** and its **subsites** with a default set of role definitions, as specified by input parameters. If the specified site (2) inherits its security
permissions, then a new security scope will be made to hold the new security settings. The procedure `proc_SecResetWebToDefaultRoleDefinition` will fail if the specified site (2) does not exist or if the specified site (2) has a unique role definition.

```sql
PROCEDURE proc_SecResetWebToDefaultRoleDefinition(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @AuthorID                      int,
    @AdminsName                    nvarchar(255),
    @AdminsDescription             nvarchar(512),
    @AdminsPermMask                bigint,
    @AuthorsName                   nvarchar(255),
    @AuthorsDescription            nvarchar(512),
    @AuthorsPermMask               bigint,
    @ContributorsName              nvarchar(255),
    @ContributorsDescription       nvarchar(512),
    @ContributorsPermMask          bigint,
    @BrowsersName                  nvarchar(255),
    @BrowsersDescription           nvarchar(512),
    @BrowsersPermMask              bigint,
    @GuestsName                    nvarchar(255),
    @GuestsDescription             nvarchar(512),
    @GuestsPermMask                bigint,
    @AnonymousPermMask             bigint,
    @ScopeId                       uniqueidentifier OUTPUT,
    @RequestGuid                   uniqueidentifier   = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2).

@AuthorID: The user identifier of the principal (1) to be assigned to the Administrator Role. This MUST be the user identifier of an existing principal in the site collection.

@AdminsName: The display name for the Administrator Role. This value MUST NOT be NULL.

@AdminsDescription: The description for the Administrator Role.

@AdminsPermMask: The WSS Rights Mask (section 2.2.2.15) for the Administrator Role. This value MUST NOT be NULL.

@AuthorsName: The display name for the Web Designer Role. This value MUST NOT be NULL.

@AuthorsDescription: The description for the Web Designer Role.

@AuthorsPermMask: The WSS Rights Mask for the Web Designer Role. This value MUST NOT be NULL.

@ContributorsName: The display name for the Contributor Role. This value MUST NOT be NULL.

@ContributorsDescription: The description for the Contributor Role.

@ContributorsPermMask: The WSS Rights Mask for the Contributor Role. This value MUST NOT be NULL.

@BrowsersName: The display name for the Reader Role. This value MUST NOT be NULL.

@BrowsersDescription: The description for the Reader Role.

@BrowsersPermMask: The WSS Rights Mask for the Reader Role. This value MUST NOT be NULL.

@GuestsName: The display name for the Guest Role. This value MUST NOT be NULL.
@GuestsDescription: The description for the Guest Role.

@GuestsPermMask: The WSS Rights Mask for the Guest Role. This value MUST NOT be NULL.

@AnonymousPermMask: A WSS Rights Mask which indicates the rights granted to a User that is anonymous, or has no specific rights.

@ScopeId: The scope identifier of the security scope of the updated permissions, returned as an output parameter.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecResetWebToDefaultRoleDefinition stored procedure returns an integer return code which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>Site does not exist or Site has a unique role definition associated with it.</td>
</tr>
</tbody>
</table>

If execution is not successful, then the proc_SecResetWebToDefaultRoleDefinition stored procedure MUST return no result sets. Otherwise, proc_SecResetWebToDefaultRoleDefinition MUST return the Site Audit Mask Result Set (section 3.1.5.113.1), repeated five times.

3.1.5.113.1 Site Audit Mask Result Set

The Site Audit Mask Result Set is returned six times upon successful execution. The first is returned after the Administrator Role has been added, the second after the principal specified by @AuthorId has been newly assigned to the administrator role and security scope, and then once per additional role that is added.

Each Site Audit Mask Result Set contains the Audit Flags (section 2.2.2.1) of the Site and MUST hold a single row, and is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.4.22).

3.1.5.114 proc_SecResolvePrincipal

The proc_SecResolvePrincipal stored procedure retrieves information about a principal (1) or site group in the specified site collection, given the principal's login name, e-mail address, or display name.

```sql
PROCEDURE proc_SecResolvePrincipal(
    @SiteId                        uniqueidentifier,
    @Input                         nvarchar(255),
    @InputIsEmailOnly             bit,
    @AccountName                   nvarchar(255),
    @DLAlias                       nvarchar(128),
    @DLAliasServerAddress         nvarchar(255),
    @SearchScope                   int,
    @RequestGuid                   uniqueidentifier   = NULL OUTPUT
);```

@SiteId: The Site Collection Identifier of the site collection.

@Input: The principal's display name, e-mail address, or login name. This parameter is used to find the principal.
@InputIsEmailOnly: Specifies how input parameters are used to find the principal. If this parameter is 0, then matching in the following order:

1. @Input to site group display name;
2. @AccountName or @Input (when @AccountName is NULL) as login name;
3. @Input as mobile phone number;
4. @Input or @DLAlias as email address;
5. @Input to principal's display name.

For any value of @InputIsEmailOnly that is not 0, skip steps 1, 2 and 3.

@AccountName: The principal's login name. If NULL, and @Input did not match on display name, then the value of @Input will be taken as login name.

@DLAlias: The principal's email alias to search for.

@DLAliasServerAddress: The server portion of an e-mail address. If the principal is a site group whose e-mail address is available, then this is used to construct the full e-mail address in the {Email} column in the Principal Information Result Set (section 3.1.5.114.1). If this parameter is NULL and the principal is a site group, {Email} MUST return NULL.

@SearchScope: A bit mask signifying the scope in which to search for matching principals. This can have one or more flags set. The following bits within @SearchScope can be set to 1.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>User</td>
</tr>
<tr>
<td>0x04</td>
<td>Domain group</td>
</tr>
<tr>
<td>0x08</td>
<td>Site group</td>
</tr>
</tbody>
</table>

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecResolvePrincipal stored procedure MUST return an integer return code of 0.

If a match for the principal is found, then the proc_SecResolvePrincipal stored procedure MUST return the Principal Information Result Set. Otherwise, it MUST NOT return a result set.

3.1.5.114.1 Principal Information Result Set

The Principal Information Result Set MUST return a single row containing information about the principal (1) which uniquely matches the input parameters.

{MatchType} int,
{PrincipalType} int,
{Id} int,
{Login} nvarchar(255),
{Email} nvarchar(255),
{DisplayName} nvarchar(255);

{MatchType}: Designates the type of match found between input variables and a principal in the site group. Values MUST be listed in the following table.
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Matched by login name.</td>
</tr>
<tr>
<td>2</td>
<td>Matched by e-mail address.</td>
</tr>
<tr>
<td>4</td>
<td>Matched by display name.</td>
</tr>
<tr>
<td>5</td>
<td>Matched by mobile phone number.</td>
</tr>
</tbody>
</table>

**{PrincipalType}**: Designates the type of the principal. Value MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Domain User.</td>
</tr>
<tr>
<td>4</td>
<td>Domain Group.</td>
</tr>
<tr>
<td>8</td>
<td>Site group.</td>
</tr>
</tbody>
</table>

**{Id}**: The identifier for the principal. If no match was found then the value is -1.

**{Login}**: The login name of the principal. This parameter is NULL if no match was found.

**{Email}**: The e-mail address, for example, someone@example.com, of the principal. This parameter is NULL if no match was found, if no e-mail address is available, or if the principal is a site group but @DLAliasServerAddress was NULL.

**{DisplayName}**: The display name of the principal. The value is NULL if no match was found.

### 3.1.5.115 proc_SecSetSiteGroupProperties

The **proc_SecSetSiteGroupProperties** stored procedure is invoked to update properties of a site (2) group.

```sql
PROCEDURE proc_SecSetSiteGroupProperties(
    @SiteId uniqueidentifier,
    @GroupId int,
    @Title nvarchar(255),
    @Description nvarchar(512),
    @OwnerID int,
    @OwnerIsUser bit,
    @UserID int,
    @SiteAdmin bit,
    @GroupOwnerID int,
    @CurrentUserIsOwner bit,
    @DLAlias nvarchar(255),
    @DLErrorMessage nvarchar(255),
    @DLFlags int,
    @DLJobId int,
    @DLArchives varchar(4000),
    @RequestEmail nvarchar(255),
    @Flags int,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

**@SiteId**: The Site Collection Identifier (section 2.2.1.1.9) of the site (2) group to update.

**@GroupId**: The identifier for the site (2) group to update.
@Title: The new user-friendly display name for the site (2) group. If this parameter is NULL, the update MUST fail and the return code MUST be 80. However, if the user does not have sufficient permissions, the return code MUST be five.

@Description: The new description for the site (2) group. This parameter can be NULL.

@OwnerID: The identifier of a user, domain group, or site (2) group to be set as the owner of the site (2) group. This parameter MUST contain the identifier of an existing user, domain group, or site (2) group.

@OwnerIsUser: A bit which specifies whether the value in @OwnerID is a user or domain group rather than a site (2) group. A value of one specifies that the value in @OwnerID is a user or domain group. A value of "0" specifies that the value in @OwnerID is a site (2) group identifier.

@UserID: The User Identifier (section 2.2.1.1.13) of the current user. This parameter MUST be ignored by proc_SecSetSiteGroupProperties.

@SiteAdmin: A bit flag specifying whether the current user has site collection administrator permission level. A value of one specifies that the current user has site collection administrator permission level. This parameter is used to check user's permission to change the site (2) group permissions and it can be NULL.

@GroupOwnerId: The User Identifier or Site Group Identifier (section 2.2.1.1.10) to compare with the existing owner of the site (2) group. Site (2) group management permissions MUST be granted if the value in @GroupOwnerId matches the identifier of the existing owner of the site (2) group and the value of @CurrentUserIsOwner is "1". This parameter is used to check user's permission to change the site (2) group permissions and it can be NULL.

@CurrentUserIsOwner: A bit flag specifying whether the value in @GroupOwnerId is to be compared with the owner of the site (2) group. Site (2) group management permissions MUST be granted if the value in @GroupOwnerId matches the identifier of the existing owner of the site (2) group and the value of @CurrentUserIsOwner is "1". This parameter can be NULL. If this parameter is NULL, the update MUST fail and the return code MUST be 80. However, if the user does not have sufficient permissions, the return code MUST be five.

@DLAlias: The new e-mail address for the distribution list associated with this site (2) group. This parameter can be NULL.

@DLErrorMessage: The most recent error message returned by an asynchronous email distribution list operation, if any. This parameter can be NULL.

@DLFlags: An integer containing new status flags for the distribution list associated with this group, as defined for the DLFlags column in the Sec_SiteGroupsView (section 2.2.6.5). This parameter can be NULL.

@DLJobId: Contains the job identifier of a new currently pending asynchronous distribution list (1) operation. A value of zero or NULL specifies there is no pending operation. This parameter can be NULL.

@DLArchives: An array of bytes containing a new list (1) of pairs of site identifiers and list identifiers defining additional lists (1), which are the recipients of e-mail sent to the distribution list via the e-mail address in @DLAlias. Each pair MUST be stored as a string, with commas separating the list's (1) parent site (2) document identifier and the list identifier, and with semicolons following each pair. This parameter can be NULL (this will overwrite the current value).

@RequestEmail: The new e-mail address used to send a request to join or depart this site (2) group. This parameter can be NULL (this will overwrite the current value).

@Flags: Contains the new membership permissions bit flags for the site (2) group. The bit flags are defined in the description of the Flags column of the Sec_SiteGroupsView table.
@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecSetSiteGroupProperties stored procedure MUST return an integer return code from the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>5</td>
<td>The current user does not have sufficient permissions to manage the site (2) group, or the site collection is set to read-only.</td>
</tr>
<tr>
<td>80</td>
<td>An update error occurred or a parameter was invalid.</td>
</tr>
</tbody>
</table>

The proc_SecSetSiteGroupProperties stored procedure MUST NOT return a result set.

### 3.1.5.116 proc_SecSetUserAccountDirectoryPath

The proc_SecSetUserAccountDirectoryPath stored procedure is invoked to set or clear the user account directory path for a specified site collection.

```sql
PROCEDURE proc_SecSetUserAccountDirectoryPath(
    @SiteId                        uniqueidentifier,
    @Restriction                   nvarchar(512),
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection.

@Restriction: Specifies the user account directory path for the specified site collection. If @Restriction is not NULL the proc_SecSetUserAccountDirectoryPath stored procedure MUST set the user account directory path for the site collection, and also set the 0x00080000 bit of the Site Collection Flags to 1. If @Restriction is NULL, then the proc_SecSetUserAccountDirectoryPath stored procedure MUST set the user account directory path for the site collection to NULL, and also set the 0x00080000 bit of the Site Collection Flags to 0.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecSetUserAccountDirectoryPath stored procedure returns an integer which MUST be 0.

The proc_SecGetUserAccountDirectoryPath stored procedure MUST NOT return a result set.

### 3.1.5.117 proc_SecSetWebRequestAccess

The proc_SecSetWebRequestAccess stored procedure is invoked to set the request access email address for a specified site (2). Site access requests to the particular site (2) are routed to the specified request access email address.

```sql
PROCEDURE proc_SecSetWebRequestAccess(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @RequestAccessEmail            nvarchar(255),
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection.
@WebId: The site identifier of the site (2) to set the request access email address.

@RequestAccessEmail: The new request access email address.

@RequestGuid: The optional request identifier for the current request.

Return Values: This procedure MUST return an integer return code of 0, and MUST NOT return a result set.

### 3.1.5.118 proc_SecUpdateAnonymousPermMask

The **proc_SecUpdateAnonymousPermMask** stored procedure is invoked to modify the permissions for the anonymous user. The site MUST have a unique scope identified by @ScopeId rather than an inherited scope. If the scope specified by @SiteId, @WebId, and @ScopeId does not exist, then the change MUST NOT occur.

```sql
PROCEDURE proc_SecUpdateAnonymousPermMask(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @ScopeId                       uniqueidentifier,
    @AnonymousPermMask             bigint,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);  
```

@SiteId: The Site Collection Identifier of the site collection containing the scope.

@WebId: The Site Identifier of the site (2) associated with the scope.

@ScopeId: The Scope Identifier of the scope containing the permission set.

@AnonymousPermMask: A WSS Rights Mask (section 2.2.2.15). The scope's permissions for anonymous Users MUST be updated to this value.

@RequestGuid: The optional request identifier for the current request.

Return Values: The **proc_SecUpdateAnonymousPermMask** stored procedure returns an integer return code which MUST be 0.

The **proc_SecUpdateAnonymousPermMask** stored procedure MUST NOT return a result set.

### 3.1.5.119 proc_SecUpdateDomainGroupMapData

The **proc_SecUpdateDomainGroupMapData** stored procedure is invoked to update the domain group map cache of a specified site collection.

```sql
PROCEDURE proc_SecUpdateDomainGroupMapData(
    @SiteId                        uniqueidentifier,
    @CacheVersion                  bigint,
    @DomainGroupMapCache           varbinary(max),
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);  
```

@SiteId: The Site Collection Identifier of the site collection to update. This parameter MUST correspond to a valid site collection, and MUST NOT be NULL.

@CacheVersion: A signed 64-bit (8-byte) integer containing the cache version number. This value is derived from the value of the SecurityVersion field contained in the site collection Access Control List (ACL). See the WSS ACL (section 2.2.3.6). This parameter MUST NOT be NULL.
@DomainGroupMapCache: The domain group map value is defined in the WSS External Group Map Cache Format (section 2.2.3.7).

<table>
<thead>
<tr>
<th>Value</th>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CachedVersion</td>
<td>8</td>
<td>The current cached security version. This is the same value as @CacheVersion.</td>
</tr>
<tr>
<td>DGCount</td>
<td>4</td>
<td>Count of Domain Groups.</td>
</tr>
<tr>
<td>REPEAT</td>
<td></td>
<td>Repeat the following fields &quot;DGCount&quot; times.</td>
</tr>
<tr>
<td>DGSigSize</td>
<td>4</td>
<td>Size of the domain group signature, in bytes.</td>
</tr>
<tr>
<td>DGSig</td>
<td>DGSigSize</td>
<td>Value of the domain group signature.</td>
</tr>
<tr>
<td>PIDCount</td>
<td>4</td>
<td>Count of PIDs (principal IDs) in this domain group.</td>
</tr>
<tr>
<td>PIDs</td>
<td>4*PIDCount</td>
<td>The value of each PID.</td>
</tr>
</tbody>
</table>

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecUpdateDomainGroupMapData stored procedure MUST return an integer return code of 0.

The proc_SecUpdateDomainGroupMapData stored procedure MUST NOT return a result set.

3.1.5.120 proc_SecUpdateRoleDef

The proc_SecUpdateRoleDef stored procedure is invoked to update a role definition with a new title, description, display order, Role Definition Type (section 2.2.1.2.16), and WSS Rights Mask (section 2.2.2.15).

PROCEDURE proc_SecUpdateRoleDef(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @RoleId                        int,
    @Title                         nvarchar(255),
    @Description                   nvarchar(512),
    @Order                         int,
    @Type                          tinyint,
    @PermMask                      bigint,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);

@SiteId: The site collection identifier of the site collection containing the role definition to be updated.

@WebId: The site identifier of the site (2) containing the role definition to be updated.

@RoleId: The role identifier of the role definition to be updated.

@Title: The title of the role definition for display in the front-end Web server. This MUST be a value that does not match an existing role definition title within the site (2), and MUST NOT be NULL.

@Description: The description of the role definition for display in the front-end Web server.

@Order: An integer value specifying the relative position in which this role definition is displayed in the front-end Web server. Multiple role definitions can have the same @Order value. This value MUST NOT be NULL.

@Type: The Role Definition Type value for the updated role.
@PermMask: A **WSS Rights Mask** that specifies the rights to grant to the role definition. If this field is not NULL, then the **WSS Rights Mask** associated with the role definition MUST be set to this value and any permissions associated with this role definition in the site (2) and site collection MUST be modified. If this field is NULL, then the existing **WSS Rights Mask** associated with the role definition MUST NOT be changed.

@RequestGuid: The optional request identifier for the current request.

Return Values: The **proc_SecUpdateRoleDef** stored procedure returns an integer return code which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>80</td>
<td>The role definition could not be updated.</td>
</tr>
</tbody>
</table>

Upon successful execution, the **proc_SecUpdateRoleDef** stored procedure MUST return a single result set as follows.

### 3.1.5.120.1 Site Audit Mask Result Set

The **Site Audit Mask Result Set** contains information about the **Audit Flags** (section 2.2.2.1) associated with the site (2). If execution is successful, then the **Site Audit Mask Result Set** MUST be returned, and MUST contain one row. The **Site Audit Mask Result Set** is defined in the Common Result Sets **Site Audit Mask Result Set** (section 2.2.4.22).

### 3.1.5.121 proc_SecUpdateUser

The **proc_SecUpdateUser** stored procedure is invoked to update the display name, e-mail address, notes, or administrative privileges listed in the user information associated with a principal (1).

```sql
PROCEDURE proc_SecUpdateUser(
    @SiteId                        uniqueidentifier,
    @CurrentUserId                 int,
    @AppPrincipalId                int,
    @UserIdToUpdate                int,
    @Title                         nvarchar(255),
    @Email                         nvarchar(255),
    @Notes                         nvarchar(1023),
    @SiteAdmin                     bit,
    @Flags                         int,
    @RequestGuid                    uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The **site collection identifier** of the **site collection** containing both the current user and the principal to be updated.

@CurrentUserId: The user identifier of the current user making the update request.

@AppPrincipalId: The **app principal** identifier of the **app** making the update request. If the request is not made by an app then this MUST be 0.

@UserIdToUpdate: The user identifier of the principal whose user information is to be updated.

@Title: The display name to be set in the principal's user information. If this parameter is NULL, then the user information update MUST fail.
@Email: The e-mail address to be set in the principal's user information. If this parameter is NULL, then the user information update MUST fail.

@Notes: The notes text to be set in the principal's user information. If this parameter is NULL, then the user information update MUST fail.

@SiteAdmin: A bit specifying whether or not to set the principal as a site collection administrator.

- When this parameter is set to "1", if the current user is a site collection administrator, and the principal is not a domain group, then the principal MUST be set as a site collection administrator.
- When this parameter is set to "0", proc_SecUpdateUser MUST clear the site collection administrator flag in the user information for the principal.
- If this parameter is NULL, then proc_SecUpdateUser MUST make no changes to the site collection administrator flag.

@Flags: A flag specifying the user information flags (section 2.2.2.12) to be set in the principal's user information.

- If the current user is NOT a site collection administrator, then the value MUST be NULL.
- If the parameter is NULL, then proc_SecUpdateUser MUST make no changes to the site collection administrator flag.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecUpdateUser stored procedure returns an integer return code which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>4335</td>
<td>Cannot successfully remove the site collection administrator privilege from the principal, as no other site collection administrator was found.</td>
</tr>
</tbody>
</table>

Result Sets: The proc_SecUpdateUser stored procedure MUST NOT return a result set.

3.1.5.122 proc_SecUpdateUserActiveStatus

The proc_SecUpdateUserActiveStatus stored procedure marks a user as an active user in a site collection.

```
PROCEDURE proc_SecUpdateUserActiveStatus(
    @SiteId uniqueidentifier,
    @UserId int,
    @RequestGuid uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection containing the user.

@UserId: The user identifier for the user to be marked as active user. If the user is not found or if the user is the system account, then proc_SecUpdateUserActiveStatus MUST take no action.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_SecUpdateUserActiveStatus stored procedure returns an integer return code which MUST be 0.
The **proc_SecUpdateUserActiveStatus** stored procedure MUST return no result sets.

### 3.1.5.123 **proc_TakeOverCheckOut**

The **proc_TakeOverCheckOut** stored procedure is invoked to override checkout for a document that is checked out and has no checked-in draft or published version.

```sql
PROCEDURE proc_TakeOverCheckOut(
    @SiteId                        uniqueidentifier,
    @ListId                        uniqueidentifier,
    @DirName                       nvarchar(256),
    @LeafName                      nvarchar(128),
    @UserId                        int,
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

- **@SiteId**: The Site Collection Identifier of the site collection containing the document.
- **@ListId**: The List Identifier of the list containing the document.
- **@DirName**: The directory name of the document.
- **@LeafName**: The leaf name of the document.
- **@UserId**: The User Identifier for the current user. This value MUST refer to an existing User Identifier for the specified site collection.
- **@RequestGuid**: The optional request identifier for the current request.

Return Values: The **proc_TakeOverCheckOut** stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution, or the User specified by <strong>@UserId</strong> does not exist, or specified document is already checked out to the specified User.</td>
</tr>
<tr>
<td>2</td>
<td>File Not Found: The specified document was not found; or the document is not checked out; or the document is checked out, but another current version of the document exists.</td>
</tr>
</tbody>
</table>

The **proc_TakeOverCheckOut** stored procedure MUST NOT return any result sets.

### 3.1.5.124 **proc_UncheckoutDocument**

The **proc_UncheckoutDocument** stored procedure is invoked to release a checked out document. The current document is set to the most recent previous published or draft version, or to the version currently checked out, depending on the document state and the **proc_UncheckoutDocument** parameters.

```sql
PROCEDURE proc_UncheckoutDocument (  
    @SiteId                        uniqueidentifier,  
    @WebId                         uniqueidentifier,  
    @DirName                       nvarchar(256),  
    @LeafName                      nvarchar(128),  
    @UserId                        int,  
    @ForceUncheckout               bit,  
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);
```

The **proc_UncheckoutDocument** stored procedure MUST NOT return any result sets.
@SiteId: The Site Collection Identifier of the site collection containing the document to have its short-term lock released or its check out reverted.

@WebId: The Site Identifier of the site (2) containing the document.

@DirName: The Document’s directory name.

@LeafName: The Document’s leaf name.

@UserId: The user identifier of the current user.

@ForceUncheckout: Specifies whether to force the reversion of a check out of the document held by another User. If this parameter is set to "1", then the checkout of the document MUST be released, and the most recent previous draft or published version MUST be reverted to the current version, even if the current user is not the user the document is checked out to. The owner of the checkout can also force the reversion of a checkout. If @ForceUncheckout is not set to "1", then the check-out of the document will not be released, and the most recent previous draft or published version will not revert to the current version if the current user is not the user the document is checked out to.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_UncheckoutDocument stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The document was not found at the specified location.</td>
</tr>
<tr>
<td>33</td>
<td>The document is long-term checked out by another user.</td>
</tr>
<tr>
<td>158</td>
<td>The document is not checked out, or does not have a short-term lock, or the current user attempted to release a short-term lock on a document that is checked out, or to revert the checkout of a document that has a short-term lock.</td>
</tr>
<tr>
<td>167</td>
<td>The document is long-term checked out and the current user attempted to release a short-term lock.</td>
</tr>
<tr>
<td>173</td>
<td>Another User has checked out the document or has a short-term lock on the document, and @Force is not set to &quot;1&quot;.</td>
</tr>
<tr>
<td>2401</td>
<td>A checked-out document was specified and no published or draft version of the document exists to revert to.</td>
</tr>
<tr>
<td>6002</td>
<td>The current user attempted to release a shared-lock.</td>
</tr>
<tr>
<td>6009</td>
<td>The current user attempted to release a lock, but the file is short-term locked by someone else.</td>
</tr>
</tbody>
</table>

proc_UncheckoutDocument MUST return 2 or 3 result sets in the order specified in the following sections.

### 3.1.5.124.1 Link Info Result Set

The Link Info Result Set returns information about all forward links and backward links associated with the document. The Link Info Result Set MUST be returned and MUST contain one row for each forward link and backward link associated with the specified document after the short-term lock has been released or the checkout has been reverted.

The Link Info Result Set is defined in the Common Result Sets Link Information Result Set (section 2.2.4.13).
3.1.5.124.2 Document Metadata Result Set

The Document Metadata Result Set returns the metadata for the specified document after the short-term lock has been released or the checkout has been reverted. The Document Metadata Result Set MUST be returned and MUST contain one row for the specified document.

The Document Metadata Result Set is defined in the Common Result Sets Document Metadata Result Set (section 2.2.4.8).

3.1.5.124.3 NULL Result Set

The NULL Result Set returns no data. The NULL Result Set MUST only be returned if the Document Metadata Result Set is not empty and @WebId is NULL. The NULL Result Set MUST return zero rows in a schema containing a single NULL column.

3.1.5.124.4 Event Receivers Result Set

The Event Receivers Result Set contains information about the list item event receivers defined for this document.

The Event Receivers Result Set MUST only be returned if the Document Metadata Result Set is not empty and @WebId is not NULL. There MUST be one row for each event receiver with an Event Host Type (section 2.2.1.2.5) of 3 (list item) registered for the specified document after the short-term lock has been released or the checkout has been reverted.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.4.11).

3.1.5.125 proc_UpdateDocBuildDependencySet

The proc_UpdateDocBuildDependencySet stored procedure is invoked to store an updated version of a build dependency set for a specified document.

PROCEDURE[proc_UpdateDocBuildDependencySet](
    @DocSiteId                     uniqueidentifier,
    @DocDirName                    nvarchar(256),
    @DocLeafName                   varchar(128),
    @Level                         tinyint,
    @BuildDependencySet           varbinary(max),
    @RequestGuid                   uniqueidentifier = NULL OUTPUT
);

@DocSiteID: The Site Collection Identifier of the site collection containing the document.

@DocDirName: The directory name of the document.

@DocLeafName: The leaf name of the document.

@Level: The Publishing Level Type value of the document containing the dependency set to update.

@BuildDependencySet: The build dependency set for the document. This parameter MUST NOT be NULL. Use proc_DeleteDocBuildDependencySet to set the document dependencies to NULL.

@RequestGuid: The optional request identifier for the current request

Return Values: The proc_UpdateDocBuildDependencySet stored procedure MUST return an integer return code of 0.
The *proc_UpdateDocBuildDependencySet* stored procedure MUST NOT return a result set.

### 3.1.5.126 proc_UpdateDocument

The *proc_UpdateDocument* stored procedure is invoked to update the metadata and *stream* of a document.

```sql
PROCEDURE proc_UpdateDocument (  
    @DocSiteId                     uniqueidentifier,  
    @DocWebId                      uniqueidentifier,  
    @DocDirName                    nvarchar(256),  
    @DocLeafName                   nvarchar(128),  
    @SendingContent                bit,  
    @DocMetaInfo                   varbinary(max),  
    @DocSize                       int,  
    @DocMetaInfoSize               int,  
    @DocFileFormatMetaInfo         varbinary(max),  
    @DocFileFormatMetaInfoSize     int,  
    @DocDirty                      bit,  
    @DocVersion                    int,  
    @DocMetaInfoVersion            int,  
    @DocFlags                      int,  
    @DocIncomingCreatedDTM         datetime,  
    @DocIncomingDTM                datetime,  
    @ThicketMainFile               bit,  
    @GetWebListForNormalization    bit,  
    @VersioningEnabled             bit,  
    @EnableMinorVersions           bit,  
    @UserId                        int,  
    @AttachmentOp                  int,  
    @PutFlags                      int,  
    @UpdateFlags                   int,  
    @CharSet                       int,  
    @ProgId                        nvarchar(255),  
    @AuditMask                     bit,  
    @WebParts                      bit,  
    @Comment                       nvarchar(1023),  
    @IaModerate                    bit,  
    @NeedsDraftOwnerRestriction    bit,  
    @WelcomePageUrl                nvarchar(260),  
    @WelcomePageParameters         nvarchar(max),  
    @VirusVendorID                 int,  
    @VirusStatus                   int,  
    @VirusInfo                     nvarchar(255),  
    @VirusInfoEx                   varbinary(max),  
    @LockTimeout                   int,  
    @RefreshLock                   bit,  
    @SharedLock                    bit,  
    @FileFragmentPartitionToCheck  tinyint,  
    @FileFragmentIdToCheck         bigint,  
    @FileFragmentPartitionsToDelete nvarchar(max),  
    @PartitionsToDelete            nvarchar(max),  
    @DocContentVerBump             int,  
    @NextBSNCHECK                  bigint,  
    @BSNBump                       bigint,  
    @StreamSchema                  tinyint,  
    @DocId                         uniqueidentifier,  
    @Level                         tinyint,  
    @DoclibRowId                   int,  
    @DocDTM                        datetime,  
    @DocUIVersion                  int,  
    @DocFlagsOut                   int,  
    @DocVersionOut                 int,  
    @RequestGuid                   uniqueidentifier = NULL  
);
@DocSiteId: The site collection identifier for the site collection containing the document to be updated.

@DocWebId: The site identifier for the site (2) containing the document.

@DocDirName: The directory name of the document location.

@DocLeafName: The leaf name of the document location.

@SendingContent: Specifies whether or not to store the document stream in the back-end database server. This attribute MUST be one if the document stream of the document is intended to be stored in the back-end database server; otherwise, it MUST be zero.

@DocMetaInfo: The metadata information for the document to be stored. If there is no metadata information for this document, this parameter MUST be NULL.

@DocSize: Final size in bytes of the document stream of the document. This MUST be zero if @SendingContent is zero.

@DocMetaInfoSize: Size in bytes of the document's metadata. This MUST be zero if @DocMetaInfo is NULL.

@DocFileFormatMetaInfo: The file format metadata stream for the document.

@DocFileFormatMetaInfoSize: Size in bytes of the document's file format metadata.

@DocDirty: A bit flag specifying whether the document has dependencies, such as links to other items, which need to be updated. If this parameter is set to "1", the document has dependencies to be updated. This flag MUST be stored by the back-end database server for subsequent updating of the dependent information.

@DocVersion: The internal version number of the document content to check on update. If this value is NULL, or the current internal version number value is NULL, or this value does not equal the current internal version number value of the document content, the entire document MUST NOT be updated. If the current internal version number value is NULL or @PutFlags contains 0x00800000 (from Put Flags Type (section 2.2.2.7)), the current internal version number MUST NOT be incremented. Otherwise, it MUST be incremented by an implementation-specific amount.

@DocMetaInfoVersion: The internal version number of the document metadata to check on update. If this value is NULL or does not equal the current internal version number value of the document metadata, the entire document MUST NOT be updated. If the document was updated, this value MUST be incremented by an implementation-specific amount.

@DocFlags: A document flag value with metadata describing the document to be updated.

@DocIncomingCreatedDTM: The datetime in UTC of the creation time stamp of the document. If this value is null, the current creation date of the document MUST NOT be updated.

@DocIncomingDTM: The datetime in UTC of the current modification time stamp of the document. If this value is null, the current last modification time stamp of the document MUST be set to the current UTC datetime of the back-end database server.

@ThicketMainFile: Specifies whether the document is a thicket main file. If this parameter is set to "1", the document is a thicket main file. If this parameter is set to "0", the document is not a thicket main file or a thicket supporting file. If this parameter is NULL, this document is a thicket supporting file.

@GetWebListForNormalization: If this parameter is set to "1", upon successful completion the procedure MUST return list (1) of sites (2) in a Site List For Normalization Result Set (section 3.1.5.126.1) whose immediate parent site is the site (2) containing the document to be updated.
@VersioningEnabled: A bit flag specifying whether historical versioning is enabled for the list (1) that contains this document. If this parameter is set to "0", a new historical version MUST NOT be created as part of this update. This parameter MUST be ignored for uncustomized items and documents that are not list items or in a document library.

@EnableMinorVersions: Specifies whether minor versions are enabled on the document. If this parameter is set to "1", minor versions MUST be enabled; otherwise, minor versions MUST NOT be enabled. This parameter MUST be ignored for uncustomized items and documents that are not list items or in a document library.

@UserId: The user identifier of the current user that is making the request to the front-end Web server.

@AttachmentOp: An integer containing Attachments Flags (section 2.2.1.2.1) describing the document.

@PutFlags: A Put Flags Type value that specifies document update options.

@UpdateFlags: A bit field that tracks additional document change options. If this parameter is "1", links that are no longer applicable after the document is updated MUST be deleted. If this parameter is not "1", it MUST be ignored.

@CharSet: An identifier for the code page to associate with the document. If this value is NULL, there MUST be no code page associated with the document.

@ProgId: Specifies a preferred application to open this document. The ProgId is used to distinguish among different applications that save files with a given file extension (for example, different editors for HTML or XML files). If this value is NULL, there MUST be no preferred application for this document.

@AuditMask: A bit flag specifying whether to return the current Audit Flags (section 2.2.1.2.1) values for the document. If this parameter is set to "1", the Site Audit Mask Result Set (section 3.1.5.126.2) MUST be returned on successful completion.

@WebParts: A bit flag specifying whether or not the update to the document includes updates to Web Parts. If this parameter is set to "1", any links to Web Parts contained inside the document MUST be removed from the back-end database server.

@Comment: A description provided when a document is checked in or published, which could be displayed by a version management UI. This value MUST be updated for the document even if it is not currently checked out.

@IsModerate: A bit flag specifying whether moderation is in effect for this document. If this parameter is set to "1", moderation is in effect on this document and moderation rules MUST be applied to prevent document publication and to implement an approval process when the document is updated.

@NeedsDraftOwnerRestriction: A bit flag specifying whether or not the draft owner for this document is the only user allowed to update it. If this parameter is set to "1" and a draft owner exists for the document, the document MUST only be updated if @UserId is the draft owner.

@WelcomePageUrl: Specifies a welcome page to redirect to when the document is fetched. If this value is NULL, there MUST be no redirection.

@WelcomePageParameters: Specifies query string or bookmark parameters, if any, to append to the URI specified by @WelcomePageUrl. If this value is NULL, there MUST be no parameters added.

@VirusVendorID: If provided, specifies the vendor identifier of the virus scanner that processed this document.

@VirusStatus: A Virus Status value (section 2.2.1.2.17) specifying the current virus check status of this document.
@VirusInfo: A string containing a provider-specific message returned by the virus scanner when it last processed the document. This value MUST be NULL, if the document does not exist or if the document has not been processed by a virus scanner.

@VirusInfoEx: A binary value that specifies extensible virus information for this document, which is opaque to the back-end server.

@LockTimeout: An integer value specifying the number of minutes a short-term lock is to be maintained for the document. A value of zero means the short-term lock MUST be released immediately. This value MUST be NULL if no short-term lock is being applied, refreshed, or released on the document.

@RefreshLock: Specifies whether to refresh a short-term lock on the document. If @LockTimeout is NULL, this parameter MUST be ignored.

- If this parameter is set to "1", the existing short-term lock on the document MUST be set to remain for the number of minutes specified by @LockTimeout.
- If this parameter is "0", the time remaining for an existing short-term lock on the document MUST remain unchanged.
- If there is no short-term lock on the document, this parameter MUST be NULL.

@SharedLock: Specifies whether the short-term lock to obtain, refresh, or release is of type shared or exclusive. If this parameter is "1", the lock type MUST be shared. Otherwise, the lock type MUST be exclusive. If @LockTimeout is NULL, this parameter MUST be ignored.

@FileFragmentPartitionToCheck: Specifies the implementation-specific file fragment partition identifier used with @FileFragmentIdToCheck.

@FileFragmentIdToCheck: Specifies the implementation-specific file fragment identifier to use for checking the implementation-specific file fragments, along with @FileFragmentPartitionToCheck. If @FileFragmentPartitionToCheck or @FileFragmentIdToCheck is NULL or the largest implementation-specific file fragment identifier in the partition specified by @FileFragmentPartitionToCheck is equal to @FileFragmentIdToCheck, the document MUST be updated and the current internal version number MUST be incremented. Otherwise it MUST NOT be incremented.

@FileFragmentPartitionsToDelete: A string of comma-separated integers that specifies which implementation-specific file fragment partition identifier to delete. If @SystemUpdate is one or @SendingContent is zero or @VirusStatus is not NULL and not three, this parameter MUST be ignored. Otherwise, the implementation-specific file fragment in those partitions MUST be deleted. The format MUST be as follows for file-fragment-partition-list.

    file-fragment-partition-list = *(""," file-fragment-partition)
    file-fragment-partition = *DIGIT

@PartitionsToDelete: A string of comma-separated integers that specifies which stream partition has all of its stream binary pieces deleted. If @SystemUpdate is one or @SendingContent is zero or @VirusStatus is not NULL and not three, this parameter MUST be ignored. Otherwise, the stream binary piece specified by the stream partitions MUST be deleted. The format MUST be the same as file-fragment-partition-list.

@DocContentVerBump: An integer value used to increase the version of the content stored in the back-end database server. This value is added to the current content version number only if @SendingContent is set to one. If @SendingContent is set to zero, this parameter MUST be ignored.

@NextBSNCheck: Specifies the current BLOB sequence number to check for the document. If this parameter is NULL or the current BLOB sequence number of the document is equal to
@NextBSNCheck, the document MUST be updated and the current internal version number MUST be incremented. Otherwise, the current internal version number MUST NOT be incremented.

@BSNBump: Specifies the amount to increase the current BLOB sequence number of the document. If @SendingContent is one, the current BLOB sequence number MUST be set to the current BLOB sequence number + @BSNBump.

@StreamSchema: Specifies the new stream schema of the document. If this parameter is not NULL, the current stream schema of the document MUST be set to @StreamSchema.

@DocId: Output parameter. The document identifier of the updated document.

@Level: Output parameter. A Publishing Level Type value (section 2.2.2.6) indicating the maximum publishing level of the updated document to be returned to the front-end Web server if multiple publishing levels of the document are available.

@DoclibRowId: Output parameter. The identifier of the list item representing the updated document. This value MUST be NULL if the document is not stored in a list (1).

@DocDTM: Output parameter. The datetime in UTC of the last modification time stamp of the updated document. If @DocIncomingDTM is NULL, this value MUST be set to the current UTC datetime according to the back-end database server; otherwise, the value MUST be set to @DocIncomingDTM.

@DocUIVersion: Output parameter. The UI version number associated with the updated document. This value MUST be NULL in the case of a document that does not exist.

@DocFlagsOut: Output parameter. A Doc Flags value (section 2.2.2.3), set to the document flag value of the updated document.

@DocVersionOut: Output parameter. The internal version counter value of the updated document. If the document was not updated, this parameter MUST be set to NULL.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_UpdateDocument stored procedure returns an integer return code, which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The document does not exist or has been deleted.</td>
</tr>
<tr>
<td>5</td>
<td>The user accessing the document is not the owner of the draft, and ownership restriction of the draft is being enforced.</td>
</tr>
<tr>
<td>30</td>
<td>An IO error occurred while deleting implementation-specific file fragments.</td>
</tr>
<tr>
<td>33</td>
<td>Locking violation or mismatch of @RefreshLock and short-term lock status before lock timeout.</td>
</tr>
<tr>
<td>80</td>
<td>This document doesn't have a document stream, but a content file exists.</td>
</tr>
<tr>
<td>87</td>
<td>The list item for this document could not be added.</td>
</tr>
<tr>
<td>154</td>
<td>The Minor UI version number has exceeded its limit.</td>
</tr>
<tr>
<td>158</td>
<td>Checkout is enforced, but the document is not checked out.</td>
</tr>
<tr>
<td>160</td>
<td>Document is at draft Level and the user accessing the document is not permitted to see drafts or is not the draft owner.</td>
</tr>
<tr>
<td>173</td>
<td>User does not have permissions for the type of checkout desired.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>212</td>
<td>The document is locked (either not the current version and is currently checked out or not enough lock quota); the document cannot be updated.</td>
</tr>
<tr>
<td>288</td>
<td>The document is not the current version; the current user is not the owner; the document cannot be updated.</td>
</tr>
<tr>
<td>1150</td>
<td>Internal Version number mismatch prevented document update.</td>
</tr>
<tr>
<td>1630</td>
<td>Unsupported file type.</td>
</tr>
<tr>
<td>1816</td>
<td>The site collection does not have enough quota to update the document.</td>
</tr>
<tr>
<td>6002</td>
<td>A short-term lock could not be obtained or modified because a shared short-term lock already exists. A shared short-term lock is used for read operations that do not change or update data, such as a SELECT statement.</td>
</tr>
<tr>
<td>6009</td>
<td>A short-term lock could not be obtained or modified because an exclusive short-term lock already exists. An exclusive short-term lock is used for data-modification operations, such as INSERT, UPDATE, or DELETE. Such a lock ensures that multiple updates cannot be made to the same resource at the same time.</td>
</tr>
</tbody>
</table>

The **proc_UpdateDocument** stored procedure can return zero to three result sets in the order listed in the following sections.

### 3.1.5.126.1 Site List For Normalization Result Set

The **Site List For Normalization Result Set** returns a list of URLs for the immediate child Sites of the Site containing the newly updated document. When the input parameter `@GetWebListForNormalization` is set to "1" and execution has been successful up to the point of updating the document, the **Site List For Normalization Result Set** MUST be returned. The **Site List For Normalization Result Set** MUST contain one row for each subsite found.

The **Site List For Normalization Result Set** is defined in the Common Result Sets **URL Result Set** (section 2.2.4.27).

### 3.1.5.126.2 Site Audit Mask Result Set

The **Site Audit Mask Result Set** returns audit configuration information. When the input parameter `@AuditMask` is set to "1", the **Site Audit Mask Result Set** MUST be returned, and MUST contain only one row. The **Site Audit Mask Result Set** is defined in the Common Result Sets **Site Audit Mask Result Set** (section 2.2.4.22).

### 3.1.5.126.3 Lock Information Result Set

The **Lock Information Result Set** returns short-term lock information for the document. The **Lock Information Result Set** MUST be returned when a successful document update has been completed, and a document has either been checked in, remains checked out, has been locked, or remains locked. The **Lock Information Result Set** MUST NOT be returned if the document has not been successfully updated or `@LockTimeout` is NULL. The **Lock Information Result Set** MUST contain only one row.

```
    tp_Login               nvarchar(255),
    CheckoutDate          datetime,
    {CheckoutExpires}     datetime;
```

**tp_Login**: The login name of the user to whom the document is checked out. If the document is currently checked in, then this MUST be NULL.
CheckoutDate: A datetime in UTC indicating when this document was checked out. If the document is currently checked in, then this MUST be NULL.

{CheckoutExpires}: A datetime in UTC indicating when the short-term lock for this document will expire. If the document is currently checked in or has a long term checkout, then this MUST be NULL.

3.1.5.127   proc_UpdateListItem

The proc_UpdateListItem stored procedure is invoked to update a list item in a document library or list.

PROCEDURE proc_UpdateListItem(
    @SiteId                        uniqueidentifier,
    @WebId                         uniqueidentifier,
    @ListId                        uniqueidentifier,
    @ItemID                        int,
    @RowOrdinal                    tinyint,
    @Size                          int,
    @ExtraItemSize                 int = NULL,
    @ItemName                      nvarchar(255)      = NULL,
    @UseNvarchar1ItemName          bit = 1,
    @ItemDirName                   nvarchar(256)      = NULL,
    @ItemLeafName                  nvarchar(256)      = NULL,
    @UserId                        int,
    @Level                         tinyint = 1,    OUTPUT,    
    @TimeNow                       datetime,
    @NeedsAuthorRestriction        bit = 0,
    @NeedsDraftOwnerRestriction    bit = 0,
    @PreserveVersion               bit = 0,
    @IsMeetingsList                bit = NULL,
    @IsIssueList                   bit = NULL,
    @IsNotUserDisplayed            bit = NULL,
    @SystemUpdate                   bit = 0,
    @ChangeLevel                   bit = 0,
    @CheckInItem                   bit = 0,
    @NeedClone                     bit = 0,
    @MajorVersionsLimit            int = 0,
    @MajorMinorVersionsLimit       int = 0,
    @IsDocLib                      bit = 0,
    @CheckSchemaVersion            int = NULL,
    @SortTypeReversed              bit = NULL,
    @tp_Ordering                   varchar(512) = NULL,
    @tp_ThreadIndex                varbinary(512) = NULL,
    @tp_HasAttachment              bit = NULL,
    @tp_ModerationStatus           int = NULL,
    @tp_IsCurrent                   bit = NULL,
    @tp_ItemOrder                  float = NULL,
    @tp_Version                    int = NULL,
    @tp_InstanceID                 int = NULL,
    @tp_ContentTypeId              tContentTypeId = NULL,
    @tp_CopySource                 nvarchar(260) = NULL,
    @tp_HasCopyDestinations        bit = NULL,
    @OnRestore                     bit = 0,
    @BumpLastDelete                bit = NULL,
    @CreateItemVersion             bit = 0,
    @UIVersion                     int = NULL,
    @NewUIVersion                  int = NULL,    OUTPUT,
    @ReturnRowset                  bit = 1,
    @tp_Author                     int = NULL,
    @tp_Editor                     int = NULL,
    @tp_Created                    datetime = NULL,
    @tp_Modified                   datetime = NULL,
    @tp_WorkflowVersion            int = NULL,
    @tp_ColumnSet                  xml = NULL,
    @nvchar1                       nvarchar(255) = NULL,
    @MtgEventType                  int = NULL,
    @RecurrenceID                  datetime = NULL,
@IsDetached                    int                = NULL,
@eventData                     varbinary(max)     = NULL,
@acl                           varbinary(max)     = NULL,
@IsFirstRow                    bit                = 1,
@tp_AppAuthor                  int                = NULL,
@tp_AppEditor                  int                = NULL,
@RequestGuid                   uniqueidentifier   = NULL OUTPUT
}

@SiteId: The site collection identifier for the site collection containing the list item to be updated.

@WebId: The site identifier for the site (2) containing the list item.

@ListId: The list identifier of the List containing the list item.

@ItemId: The list identifier of the list item.

@RowOrdinal: The 0-based ordinal of the row to update in the set of rows representing this list item in the AllUserData table (section 2.2.6.2). If a list item requires multiple rows to represent it in the AllUserData table because it contains more defined data columns than will fit in a single row, then this parameter specifies which row to update with this call. If more than one row of data for the list item is to be updated, then the front-end Web server MUST call proc_UpdateListItem once for each updated row. This parameter MUST NOT be NULL.

@Size: The new size, in bytes, of the list item row to be updated. This parameter MUST NOT be NULL.

@ExtraItemSize: The size, in bytes, of the predefined SQL parameter fields in the list item row.

@ItemName: The new display name for the list item.

@UseNvarchar1ItemName: If @ItemName is NULL, then this bit flag specifies use of the content of @nvarchar1 for the new display name for the list item. If @ItemName is not NULL, then this parameter MUST be ignored.

@ItemDirName: An output parameter containing the directory name of the updated list item.

@ItemLeafName: An output parameter containing the leaf name of the updated list item.

@UserId: The user identifier for the current user. This parameter is used for comparison when the @NeedsAuthorRestriction or @NeedsDraftOwnerRestriction parameters are set. If this parameter is NULL, then the user identifier stored as the list item's editor MUST be used as the current user.

@Level: A Publishing Level Type value specifying the publishing status of the updated list item.

@TimeNow: The current datetime in UTC. This parameter MUST NOT be NULL.

@NeedsAuthorRestriction: A bit flag specifying whether or not only the list item's author is permitted to update the list item. If this parameter is set to 1, then the current user MUST be the list item's author.

@NeedsDraftOwnerRestriction: A bit flag specifying whether or not only the list item's draft owner is permitted to update the list item. If this parameter is set to 1, the current user MUST be the list item's draft owner.

@PreserveVersion: A bit flag specifying whether to preserve the internal version number of the list item to be updated. If this parameter is set to 1, then the internal version number of the list item MUST NOT be incremented as part of the update. Otherwise, if the list item is updated, then the current internal version number MUST be incremented by an implementation-specific amount.

@IsMeetingsList: A bit flag specifying whether the list item is contained in a Meetings List (a List with a List Server Template value of 200).
@IsIssueList: A bit flag specifying whether the list item is contained in an Issues List (a List with a **List Server Template** value of 1100). This parameter MUST be ignored.

@IsNotUserDisplayed: A bit flag specifying whether the display name of the current user is not to be displayed in logging events (1) for this update. If this parameter is set to 1, then the display name of the current user MUST NOT be used when **proc_UpdateListItem** logs events (1) because of the update, and the string "***" MUST be used instead.

@SystemUpdate: A bit flag specifying whether to leave the last modification time stamp and the list item editor unchanged during this update. If this parameter is set to 1, then the list item MUST be updated without affecting the current last modification time or list item editor.

@ChangeLevel: A bit flag specifying whether to recalculate the publishing level of the list item as part of the update. If this parameter is set to 1, then the list item's publishing level MUST be recalculated.

@CheckinItem: A bit flag specifying whether to set the publishing level for the updated list item to published. If this parameter is set to 1, then **proc_UpdateListItem** MUST set the updated List Item's publishing level to 1 (published).

@NeedClone: A bit flag specifying whether a copy of the list item as a draft version is made as part of this update. If this parameter is set to 1, then the publishing level specified by the @Level parameter MUST also be set to 1 (published).

@MajorVersionsLimit: The number of major versions to keep of the list item.

@MajorMinorVersionsLimit: The total combined number of major versions and minor versions to keep of the list item.

@IsDocLib: A bit flag specifying whether the list item to be updated is contained within a document library. If this parameter is set to 1, then the List specified by @ListId MUST be a document library.

@CheckSchemaVersion: This specifies a list schema version number to compare with the list schema version of the list item's containing List. If this parameter is not NULL and does not match the current list schema version number, then the list item MUST NOT be updated.

@SortTypeReversed: If this parameter is 1, then the sort behavior for this item MUST be changed to the opposite of the value of the list item's type. If this parameter is 0 or NULL, then the sort behavior for this item MUST NOT be changed.

@tp_Ordering: This specifies the threading structure for this list item in a legacy Discussion Board List (a list with a **List Base Type** (section 2.2.1.2.11) of 3) as a concatenation of timestamp values in yyyyMMddHHmmss format. For all list items in lists with other **List Base Types**, this parameter MUST be NULL.

@tp_ThreadIndex: This specifies the list item's position within a threaded legacy Discussion Board List (a List with a **List Base Type** of 3) as a binary structure. For all List Items in Lists with other **List Base Types**, this parameter MUST be NULL.

@tp_HasAttachment: A bit flag specifying whether the list item has an associated attachment.

@tp_ModerationStatus: A Moderation Status value specifying the current moderation approval status of this list item.

@tp_IsCurrent: A bit flag specifying whether this is the current version of this publishing level of the list item.

@tp_ItemOrder: This specifies the implementation-specific order in which the list item is displayed with other list items from the same list. This value can be the same as other list items in the list.
@tp_Version: The internal version number value of the list item to update. If this parameter is not NULL and does not equal the current value of the internal version number of the list item, then the item MUST NOT be updated.

@tp_InstanceID: This parameter MUST be ignored.

@tp_ContentTypeId: This specifies the Content Type Identifier of the content type for this list item. For details on the tContentTypeId custom type, see section 2.2.1.1.1.

@tp_CopySource: The URL used as a source for this list item. If this list item was not copied from a source list item, then this value MUST be NULL.

@tp_HasCopyDestinations: A bit flag specifying whether destination locations have been set for this list item to be copied to. If this list item does not have a destination location set, then this value MUST be 0.

@OnRestore: A bit flag specifying whether this list item is being inserted by an implementation-specific backup restore operation.

@BumpLastDelete: A bit flag specifying whether to update the tp_LastDeleted (section 2.2.4.14) property on the List specified by @ListId. If set to 0, then the tp_LastDeleted property MUST NOT be updated.

@CreateItemVersion: A bit flag specifying whether to create a new version of the list item as part of this update.

@UIVersion: The UI version number to set for the list item.

@NewUIVersion: An output parameter returning the UI version number set for the list item, which MUST be NULL if one was not assigned.

@ReturnRowset: A bit flag specifying whether to return an Item Update Result Set (section 3.1.5.127.1).

@tp_Author: The User Identifier to set as the author of the list item.

@tp_Editor: The User Identifier to set as the last editor of the list item.

@tp_Created: A datetime in UTC to set as the list item's creation time stamp. If this parameter is NULL or @SystemUpdate is "1", then the current time stamp MUST NOT be updated.

@tp_Modified: A datetime in UTC to set as the list item's last modified time stamp. If this parameter is NULL or @SystemUpdate is "1", then the current time stamp MUST NOT be updated. If this parameter is NULL and @SystemUpdate is not "1", then @TimeNow SHOULD be set as the list item's last modified time stamp.

@tp_WorkflowVersion: If this list item is part of a workflow (2), then this specifies the value to set denoting the state of this list item within that workflow (2). If this list item is not part of a workflow (2), then this value MUST be NULL.

@tp_ColumnSet: The XML fragment that specifies the values of user-defined columns in the list. The name of each element specified in this parameter MUST be one of the columns defined in the AllUserData table (see section 2.2.6.2) with the SPARSE keyword.

@nvarchar1: User-defined column in the list containing a value of type nvarchar. If the column contains no data, then the value MUST be NULL.

@MtgEventType: If @IsMeetingsList is "1", this parameter specifies the type of calendar event.

@RecurrenceID: If @IsMeetingsList is "1", this parameter specifies a particular instance of a recurring meeting.
@IsDetached: If @IsMeetingsList is "1" and this parameter is "1", it specifies that this list item is a detached item from a meeting series.

@eventData: Contains implementation-specific event (1) data significant to the front-end Web server but otherwise opaque to the back-end database server, to be stored by the back-end database server for eventual writing to a log file.

@acl: The binary serialization of the WSS ACL Format access control list for the data supplied in @eventData, to be stored with the data.

@IsFirstRow: If @RowOrdinal is 0, then this parameter MUST be 1. Otherwise this parameter MUST be 0.

@tp_AppAuthor: The app principal identifier of the app principal who created the list item.

@tp_AppEditor: The app principal identifier of the app principal who last edited the list item.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_UpdateListItem stored procedure returns an integer return code, which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>Could not create a unique filename, or the specified Site does not exist.</td>
</tr>
<tr>
<td>5</td>
<td>The current user does not have sufficient permissions to update the list item.</td>
</tr>
<tr>
<td>13</td>
<td>The list item to be added is not valid.</td>
</tr>
<tr>
<td>87</td>
<td>Unable to update the list item because the input parameters do not match an existing list item, or an error occurred during a table update operation.</td>
</tr>
<tr>
<td>160</td>
<td>A valid User Identifier was not specified.</td>
</tr>
<tr>
<td>212</td>
<td>The database for the Site is locked.</td>
</tr>
<tr>
<td>288</td>
<td>The list item cannot be published by the current user because a different User is listed as the draft owner.</td>
</tr>
<tr>
<td>1150</td>
<td>Failed to update the list item.</td>
</tr>
<tr>
<td>1638</td>
<td>The current schema version of the List does not match the value specified in @CheckSchemaVersion.</td>
</tr>
<tr>
<td>1639</td>
<td>Only a published list item can be copied.</td>
</tr>
<tr>
<td>1816</td>
<td>The site collection is over its allocated size quota.</td>
</tr>
<tr>
<td>4005</td>
<td>The specified list item was not found.</td>
</tr>
</tbody>
</table>

The proc_UpdateListItem stored procedure MUST return zero or one result sets as follows.

3.1.5.127.1  Item Update Result Set

The Item Update Result Set returns status information about the list item update. The Item Update Result Set MUST only be returned when the @ReturnRowset parameter is set to 1, and MUST contain a single row.

{ReturnCode}    int,
{ReturnCode}: The return code value returned by proc_UpdateListItem.

{RowsAffected}: The count of the AllUserData table (section 2.2.6.2) rows updated by proc_UpdateListItem.

{Version}: The current value of the internal version number for the updated list item. This MUST be NULL if the specified list item was not found.

{Author}: The User Identifier for the author of the updated list item. This MUST be NULL if the specified list item was not found.

{Id}: The List Item Identifier of the updated list item. This MUST be NULL if the specified list item was not found.

{AuditMask}: The Audit Flags (section 2.2.2.1) value for the updated list item. This MUST be NULL if the specified list item was not found.

{InheritAuditMask}: The Audit Flags value for the updated list item inherited from its containing List or document library. This MUST be NULL if the specified list item was not found.

{GlobalAuditMask}: The global Audit Flags value for the site collection specified by @SiteId. This MUST be NULL if the specified site collection was not found.

{FullUrl}: The store-relative form URL for the updated list item. This MUST be NULL if the specified list item was not found.

3.1.5.128 proc_UpdateListSettings

The proc_UpdateListSettings stored procedure is invoked to update list (1) metadata.

PROCEDURE proc_UpdateListSettings(
    @SiteId                        uniqueidentifier, 
    @WebId                         uniqueidentifier, 
    @ListId                        uniqueidentifier, 
    @BaseType                      int, 
    @ServerTemplate                int, 
    @Title                         nvarchar(255), 
    @ReadSecurity                  int, 
    @WriteSecurity                 int, 
    @NewDefaultView                uniqueidentifier, 
    @Description                   nvarchar(max), 
    @TemplateDirName               nvarchar(256), 
    @TemplateLeafName              nvarchar(128), 
    @Fields                        tCompressedString, 
    @Direction                     int, 
    @EventSinkAssembly             nvarchar(255), 
    @EventSinkClass                nvarchar(255), 
    @EventSinkData                 nvarchar(255), 
    @Flags                         bigint, 
    @ImageUrl                      nvarchar(255), 
    @ThumbnailSize                 int, 
    @WebImageWidth                 int, 
    @WebImageHeight                int, 
)
@SiteId: The site collection identifier of the site collection containing the list (1) whose metadata is being updated.

@WebId: The site identifier of the site (2) containing the list (1).

@ListId: The list identifier of the list (1) to be updated.

@BaseType: A List Base Type value (section 2.2.1.2.11). If @ConvertToGlobalMtgDataList is equal to "1", this value MUST be used to set the List Base Type; otherwise, the value MUST be ignored.

@ServerTemplate: The identifier for the List Server Template (section 2.2.1.2.12) defining the base structure of this list (1). If @ConvertToGlobalMtgDataList is equal to "1", this value MUST be used to set the List Server Template; otherwise, the value MUST be ignored.

@Title: A user-provided title text for the list (1). If this parameter is NULL, the current value MUST NOT be changed.

@ReadSecurity: Specifies the security policy that MUST be set for read access on list items in the list (1). Valid read security values are listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Users with read permissions on this list (1) can read all list items.</td>
</tr>
<tr>
<td>2</td>
<td>Users with read permissions on this list (1) can only read their own list items.</td>
</tr>
<tr>
<td>NULL</td>
<td>There MUST be no changes to read security for the list (1).</td>
</tr>
</tbody>
</table>

@WriteSecurity: Specifies the security policy that MUST be set for write access on list items in the list (1). Valid write security values are listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Users with write permissions on this list (1) can create list items and have write access to all list items.</td>
</tr>
<tr>
<td>2</td>
<td>Users with write permissions on this list (1) can create list items and have write access to their own list items only.</td>
</tr>
<tr>
<td>4</td>
<td>Users have no write access to any list items.</td>
</tr>
<tr>
<td>NULL</td>
<td>There MUST be no changes to write security for the list (1).</td>
</tr>
</tbody>
</table>
@NewDefaultView: A **View Identifier** (section 2.2.1.14) of a defined **list view** to be displayed by default when navigating to the list (1). If this parameter is NULL, the current value MUST NOT be changed.

@Description: A user-provided readable text description for the list (1). If this parameter is NULL, the current value MUST NOT be changed.

@TemplateDirName: Directory name for the **document template** associated with the list (1). This parameter can be NULL. The parameter MUST be ignored if @TemplateLeafName is NULL or is an empty string.

@TemplateLeafName: **Leaf name** for the document template associated with the list (1). If @TemplateLeafName is NULL, there MUST be no change to the current document template. If @TemplateLeafName contains an empty string, the current document template MUST be set to NULL. If @TemplateLeafName is not NULL and is not an empty string, the current document template MUST be set to the **Document Identifier** (section 2.2.1.1.2) of the document template specified by @TemplateDirName and @TemplateLeafName.

@Fields: Specifies the field definitions for the list (1), consisting of the concatenation of an implementation-specific version string for the **list template**, followed by an **XML** representation of the field definitions. If this parameter is NULL, the current value MUST NOT be changed.

@Direction: An enumerated value specifying the direction of text flow for **front-end Web server** elements presented by this list (1). If this parameter is NULL, the current value MUST NOT be changed. Otherwise, the parameter MUST be one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No explicit direction is specified.</td>
</tr>
<tr>
<td>1</td>
<td>Text flow is left to right.</td>
</tr>
<tr>
<td>2</td>
<td>Text flow is right to left.</td>
</tr>
</tbody>
</table>

@EventSinkAssembly: A .NET assembly name for an **event sink** handler for the list (1). If this parameter is NULL, the current value MUST NOT be changed. If this parameter is an empty string, the current value MUST be set to NULL.

@EventSinkClass: A .NET assembly class identifier for an event sink handler for the list (1). If this parameter is NULL, the current value MUST NOT be changed. If this parameter is an empty string, the current value MUST be set to NULL.

@EventSinkData: Data for an event sink handler for the list (1). If this parameter is NULL, the current value MUST NOT be changed. If this parameter is an empty string, the current value MUST be set to NULL.

@Flags: A **List Flags** value (section 2.2.2.5) for the list (1). If @Fields is not NULL, this value MUST be a bitwise OR with 0x800000 (LP_FIELDSCHEMAMODIFIED). If this parameter is NULL, the current value MUST NOT be changed.

@ImageUrl: A **site-relative URL** holding an image associated with the list (1). If this parameter is NULL, the current value MUST NOT be changed.

@ThumbnailSize: A value specifying the width in pixels used by lists (1) that have a **List Server Template** value (section 2.2.1.2.12) of 109 (Image Library Template) to determine the rendering size of an image thumbnail. If this parameter is NULL, the current value MUST NOT be changed.

@WebImageWidth: A value specifying the width in pixels used by lists (1) that have a **List Server Template** value of 109 (Image Library Template) to determine the rendering width of an image. If this parameter is NULL, the current value MUST NOT be changed.
@WebImageHeight: A value specifying the height in pixels used by lists (1) that have a List Server Template value of 109 (Image Library Template) to determine the rendering height of an image. If this parameter is NULL, the current value MUST NOT be changed.

@Version: An integer used to match on the internal version counter of the list (1) to be updated. This is used to prevent updates with out-of-date data. On successful execution, the internal version counter value of the list (1) will be incremented by one. If this parameter does not match the current version, the list (1) MUST NOT be updated.

@ConvertToGlobalMtgDataList: Used to convert the data of the list (1) to use global meeting data. If this parameter is set to one, the data MUST be converted to global meeting data and the List Server Template specified by @ServerTemplate MUST NOT be any of the following.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Meeting</td>
</tr>
<tr>
<td>202</td>
<td>Meeting User</td>
</tr>
<tr>
<td>212</td>
<td>Homepage</td>
</tr>
</tbody>
</table>

If set to any other value or NULL, the meeting data for this list (1) MUST NOT be changed.

@EmailAlias: The email alias of the list (1). This alias is used to allow documents and list items to be sent directly to this list (1) through an implementation-specific email-handling feature. If this parameter is NULL, the current value MUST NOT be changed.

@SendToLocation: An implementation-specific Send-To string holding a URL used for the list (1). If this parameter is NULL, the current value MUST NOT be changed.

@NavBarEid: Specifies a navigation node element identifier for the parent navigation node of the node that will represent the list (1). If @AddOrRemoveFromNavBar is NULL or zero, this parameter MUST be ignored. Otherwise, it MUST contain a valid navigation node element identifier.

@AddOrRemoveFromNavBar: Specifies whether to add or remove this list (1) from the site's top link bar. If this parameter is set to "0" and the list (1) was updated, the list (1) MUST be removed from the site's top link bar. If this parameter is set to "1" and the list (1) was updated and the list (1) is not in the site's top link bar, the list (1) MUST be added to the site's top link bar. To move nodes from one location to another, the navigation node SHOULD be first removed and then re-added with the new desired setting or else duplicate node entries will occur.

@UseRootFolderForNav: Specifies whether the root folder of the list (1) is used for the list (1) entry in the site's top link bar, instead of the default view page. If this parameter is set to "1" and the list (1) was updated, the root folder of the list (1) MUST be set as the top link bar target. If this parameter is set to "0" and the list (1) was updated, the default view page for the list (1) MUST be set as the top link bar target. If the @AddOrRemoveFromNavBar parameter is set to NULL, this parameter MUST also be NULL. Otherwise, this parameter can be NULL, which MUST have the effect of being set to "0". To change the root folder navigation from one node to another, the current navigation node SHOULD be first removed and the new one added with the desired setting or else duplicate node entries will occur.

@MajorVersionCount: Specifies the maximum number of major versions of list items to be retained by the list (1).

@MajorMinorVersionCount: Specifies the maximum number of minor versions of list items to be retained by the list (1).

@WorkFlowId: Specifies the Workflow Identifier (section 2.2.1.1.16) of the default workflow of the list (1).
@RemoveSystemAlerts: A bit flag specifying whether to remove system alerts for this list (1) from the email subscriptions for the system administration user. If this parameter is set to "1", system alerts on the list (1) will be removed.

@Followable: If this list (1) is followable, this value MUST be one. Otherwise, it MUST be zero.

@ValidationFormula: The implementation-specific validation formula to be used for items in this list (1). If this parameter is NULL, the current value MUST NOT be changed.

@ValidationMessage: The message to display to the user when the implementation-specific list (1) validation formula fails. If this parameter is NULL, the current value MUST NOT be changed.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_UpdateListSettings stored procedure returns an integer return code that MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>3</td>
<td>The site (2) does not exist or list (1) does not exist.</td>
</tr>
<tr>
<td>13</td>
<td>@ConvertToGlobalMtgDataList is set to one and @ServerTemplate is 200, 202, or 212 so list (1) cannot use global meeting data.</td>
</tr>
<tr>
<td>15</td>
<td>The document template URL (combination of @TemplateDirName and @TemplateLeafName) is not valid.</td>
</tr>
<tr>
<td>52</td>
<td>Another list (1) with @Title already exists.</td>
</tr>
<tr>
<td>1150</td>
<td>@Version did not match the current version of the list (1).</td>
</tr>
<tr>
<td>5069</td>
<td>@ReadSecurity is 0x00000002 (LSReadOnlyMyItems) and @WriteSecurity is not NULL and the list (1) contains a field with an implementation-specific list (1) validation constraint.</td>
</tr>
</tbody>
</table>

The proc_UpdateListSettings stored procedure MUST NOT return any result sets.

### 3.1.5.129 proc_UpdateUserInfoInTableFromRowUpdater

The proc_UpdateUserInfoInTableFromRowUpdater stored procedure updates the user information data for the specified user.

```sql
PROCEDURE proc_UpdateUserInfoInTableFromRowUpdater(
    @SiteId uniqueidentifier,
    @UserId int,
    @Title nvarchar(255) = NULL,
    @Email nvarchar(255) = NULL,
    @IsActive bit = NULL,
    @Locale int = NULL,
    @CalendarType smallint = NULL,
    @AdjustHijriDays smallint = NULL,
    @TimeZone smallint = NULL,
    @Collation smallint = NULL,
    @Time24 bit = NULL,
    @AltCalendarType tinyint = NULL,
    @CalendarViewOptions tinyint = NULL,
    @WorkDays smallint = NULL,
    @WorkDayStartHour smallint = NULL,
    @WorkDayEndHour smallint = NULL,
    @MobileNumber nvarchar(127) = NULL,
    @MUILanguages varchar(64) = NULL,
    @ContentLanguages varchar(64) = NULL
)
```
@SiteId: The Site Collection Identifier (section 2.2.1.1.9) of the site collection containing the user whose information is to be updated.

@UserId: The User Identifier (section 2.2.1.1.13) for the user whose information is to be updated.

@Title: The user-friendly display name of the user. If this parameter is NULL, then the display name MUST be set to the empty string.

@Email: The email address of the user. If this parameter is NULL, then the email address MUST be set to the empty string.

@IsActive: A bit flag specifying whether the user is an active user in the site collection. This flag is set to 0 if the User is not an active user. This flag is set to 1 to indicate otherwise. If this flag is NULL, then the value MUST NOT be changed.

@Locale: An LCID specifying the preferred locale settings to be used when formatting and displaying UI for the user.

@CalendarType: The Calendar Type (section 2.2.1.2.3) to be used when processing date values for this user.

@AdjustHijriDays: If the @CalendarType parameter value is 6, then this parameter specifies the number of days to extend or reduce the current month in Hijri calendars for this User.

@TimeZone: The Time Zone Identifier for the time zone to be used when displaying time values for this User.

@Collation: The Collation Order (section 2.2.1.2.4) to be used when displaying information to this User.

@Time24: A bit flag which specifies whether to use a 24-hour time format when displaying time values to this User. If this parameter is set to 1, then the 24-hour time format is used; otherwise, the 12-hour time format is used.

@AltCalendarType: The Calendar Type of an alternate calendar for processing date values for this User.

@CalendarViewOptions: A Calendar View Options Type (section 2.2.3.1) which specifies the calendar display options setting for this User.

@WorkDays: A set of Workdays Flags (section 2.2.2.14) which specify the week days defined as the work week for this User.

@WorkDayStartHour: The start time of the work day for this user, in minutes from 12:00AM. For example, the value 480 indicates 8:00AM.

@WorkDayEndHour: The end time of the work day for this user, in minutes from 12:00AM. For example, the value 1020 indicates 5:00PM.

@MobileNumber: The user-friendly text field that specifies the mobile phone number of the user.

@MUILanguages: A string that contains the distinct culture(s) for the user's preferred display language(s), separated by semicolon. This string can be NULL.

@ContentLanguages: A string that contains the distinct culture(s) for the user's preferred content language(s), separated by semicolon. This string can be NULL.
@RequestGuid: The optional request identifier for the current request.

Return Values: This procedure returns an integer return code which MUST be 0.

This procedure MUST return no result sets.

3.1.5.130  proc_UrlToWebUrl

The proc_UrlToWebUrl stored procedure returns the store-relative form URL of the site (2) that contains a specified store-relative form URL and is inside a specific site collection.

PROCEDURE proc_UrlToWebUrl(
    @WebSiteId  uniqueidentifier,
    @Url        nvarchar(260),
    @RequestGuid uniqueidentifier = NULL OUTPUT
);

@WebSiteId: The Site Collection Identifier (section 2.2.1.1.9) for a site collection that contains the @Url parameter value. For successful execution, the Site Collection Identifier MUST match an existing site collection which could hold the @Url parameter value. If the identifier does not match an existing site collection, then an integer value of 1168 MUST be returned along with a Web URL Result Set.

@Url: A store-relative URL.

@RequestGuid: The optional request identifier for the current request.

Return Values: The proc_UrlToWebUrl stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>1168</td>
<td>The site collection specified by @WebSiteId does not exist or @Url is not contained in the specified site collection.</td>
</tr>
</tbody>
</table>

The proc_UrlToWebUrl stored procedure MUST return a single result set as follows.

3.1.5.130.1  Web URL Result Set

The Web URL Result Set returns the store-relative form URL of the site (2) containing the @Url input parameter value. The Web URL Result Set MUST be returned and MUST contain one row.

{WebUrl} nvarchar(256);

{WebUrl}: The store-relative form URL of the site (2) that contains the @Url parameter value. This MUST be an empty string when the @Url parameter is invalid or is not contained in the specified site collection or the specified site collection does not exist. @Url is only checked for whether a valid subsite path exists within @Url.

3.1.5.131  proc_WriteChunkToAllDocStreams

The proc_WriteChunkToAllDocStreams stored procedure establishes new document content associated with the document specified by @DocId, or appends to already existing document content established by an earlier invocation of proc_WriteChunkToAllDocStreams.
PROCEDURE dbo.proc_WriteChunkToAllDocStreams(
    @SiteId uniqueidentifier,
    @DocId uniqueidentifier,
    @Offset int,
    @00 varbinary(max) = NULL,
    @01 varbinary(max) = NULL,
    @02 varbinary(max) = NULL,
    @03 varbinary(max) = NULL,
    @04 varbinary(max) = NULL,
    @05 varbinary(max) = NULL,
    @06 varbinary(max) = NULL,
    @07 varbinary(max) = NULL,
    @08 varbinary(max) = NULL,
    @09 varbinary(max) = NULL,
    @0A varbinary(max) = NULL,
    @0B varbinary(max) = NULL,
    @0C varbinary(max) = NULL,
    @0D varbinary(max) = NULL,
    @0E varbinary(max) = NULL,
    @0F varbinary(max) = NULL,
    @10 varbinary(max) = NULL,
    @11 varbinary(max) = NULL,
    @12 varbinary(max) = NULL,
    @13 varbinary(max) = NULL,
    @14 varbinary(max) = NULL,
    @15 varbinary(max) = NULL,
    @16 varbinary(max) = NULL,
    @17 varbinary(max) = NULL,
    @18 varbinary(max) = NULL,
    @19 varbinary(max) = NULL,
    @1A varbinary(max) = NULL,
    @1B varbinary(max) = NULL,
    @1C varbinary(max) = NULL,
    @1D varbinary(max) = NULL,
    @1E varbinary(max) = NULL,
    @1F varbinary(max) = NULL,
    @20 varbinary(max) = NULL,
    @21 varbinary(max) = NULL,
    @22 varbinary(max) = NULL,
    @23 varbinary(max) = NULL,
    @24 varbinary(max) = NULL,
    @25 varbinary(max) = NULL,
    @26 varbinary(max) = NULL,
    @27 varbinary(max) = NULL,
    @28 varbinary(max) = NULL,
    @29 varbinary(max) = NULL,
    @2A varbinary(max) = NULL,
    @2B varbinary(max) = NULL,
    @2C varbinary(max) = NULL,
    @2D varbinary(max) = NULL,
    @2E varbinary(max) = NULL,
    @2F varbinary(max) = NULL,
    @30 varbinary(max) = NULL,
    @31 varbinary(max) = NULL,
    @32 varbinary(max) = NULL,
    @33 varbinary(max) = NULL,
    @34 varbinary(max) = NULL,
    @35 varbinary(max) = NULL,
    @36 varbinary(max) = NULL,
    @37 varbinary(max) = NULL,
    @38 varbinary(max) = NULL,
    @39 varbinary(max) = NULL,
    @3A varbinary(max) = NULL,
    @3B varbinary(max) = NULL,
    @3C varbinary(max) = NULL,
    @3D varbinary(max) = NULL,
    @3E varbinary(max) = NULL,
    @3F varbinary(max) = NULL,
    @40 varbinary(max) = NULL,
@41 varbinary(max) = NULL,
@42 varbinary(max) = NULL,
@43 varbinary(max) = NULL,
@44 varbinary(max) = NULL,
@45 varbinary(max) = NULL,
@46 varbinary(max) = NULL,
@47 varbinary(max) = NULL,
@48 varbinary(max) = NULL,
@49 varbinary(max) = NULL,
@4A varbinary(max) = NULL,
@4B varbinary(max) = NULL,
@4C varbinary(max) = NULL,
@4D varbinary(max) = NULL,
@4E varbinary(max) = NULL,
@4F varbinary(max) = NULL,
@50 varbinary(max) = NULL,
@51 varbinary(max) = NULL,
@52 varbinary(max) = NULL,
@53 varbinary(max) = NULL,
@54 varbinary(max) = NULL,
@55 varbinary(max) = NULL,
@56 varbinary(max) = NULL,
@57 varbinary(max) = NULL,
@58 varbinary(max) = NULL,
@59 varbinary(max) = NULL,
@5A varbinary(max) = NULL,
@5B varbinary(max) = NULL,
@5C varbinary(max) = NULL,
@5D varbinary(max) = NULL,
@5E varbinary(max) = NULL,
@5F varbinary(max) = NULL,
@60 varbinary(max) = NULL,
@61 varbinary(max) = NULL,
@62 varbinary(max) = NULL,
@63 varbinary(max) = NULL,
@64 varbinary(max) = NULL,
@65 varbinary(max) = NULL,
@66 varbinary(max) = NULL,
@67 varbinary(max) = NULL,
@68 varbinary(max) = NULL,
@69 varbinary(max) = NULL,
@6A varbinary(max) = NULL,
@6B varbinary(max) = NULL,
@6C varbinary(max) = NULL,
@6D varbinary(max) = NULL,
@6E varbinary(max) = NULL,
@6F varbinary(max) = NULL,
@70 varbinary(max) = NULL,
@71 varbinary(max) = NULL,
@72 varbinary(max) = NULL,
@73 varbinary(max) = NULL,
@74 varbinary(max) = NULL,
@75 varbinary(max) = NULL,
@76 varbinary(max) = NULL,
@77 varbinary(max) = NULL,
@78 varbinary(max) = NULL,
@79 varbinary(max) = NULL,
@7A varbinary(max) = NULL,
@7B varbinary(max) = NULL,
@7C varbinary(max) = NULL,
@7D varbinary(max) = NULL,
@7E varbinary(max) = NULL,
@7F varbinary(max) = NULL,
@80 varbinary(max) = NULL,
@81 varbinary(max) = NULL,
@82 varbinary(max) = NULL,
@83 varbinary(max) = NULL,
@84 varbinary(max) = NULL,
@SiteId: The site collection identifier of the site collection containing the document.

@DocId: The document identifier of the document.

@Offset: This parameter MUST be zero, if no document content corresponds to @DocId; otherwise, it MUST be the length, in bytes, of the document content which corresponded to @DocId before the stored procedure was invoked.

@00 -- @FF: 256 optional parameters. The non-NULL parameters, concatenated in order according to their name, interpreted as a hexadecimal number, MUST be stored or appended to the document content. If one of the parameters is NULL, all subsequent parameters MUST be NULL.
@RequestGuid: The optional request identifier for the current request.

Return Values: The `proc_WriteChunkToAllDocStreams` stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>29</td>
<td>Unexpected error.</td>
</tr>
</tbody>
</table>

The `proc_WriteChunkToAllDocStreams` stored procedure MUST NOT return a result set.

### 3.1.5.132 proc_WriteStreamToRBS

The `proc_WriteStreamToRBS` stored procedure establishes the remote BLOB storage identifier representing a new stream binary piece for a document. For additional information regarding remote BLOB storage, see [MS-WSSO] section 2.1.2.3.8.

```sql
PROCEDURE dbo.proc_WriteStreamToRBS (
    @SiteId       uniqueidentifier,
    @DocId        uniqueidentifier,
    @Partition    tinyint,
    @BSN          bigint,
    @Size         int,
    @Type         tinyint,
    @Expiration   datetime,
    @RbsId        varbinary(64)
)
```

@SiteId: The site collection identifier of the site collection containing the document.

@DocId: The document identifier of the document.

@Partition: The stream partition that the new stream binary piece belongs to.

@BSN: The BLOB sequence number of the new stream binary piece.

@Size: The size, in bytes, of the new stream binary piece.

@Type: The stream type of the new stream binary piece.

@Expiration: The timestamp in UTC format of the stream binary piece. If the stream binary piece does not have an expiration, this parameter MUST be NULL.

@RbsId: The remote BLOB storage identifier for the new stream binary piece. For additional information regarding remote BLOB storage, see [MS-WSSO] section 2.1.2.3.8.

Return Values: The `proc_WriteStreamToRBS` stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>29</td>
<td>Unexpected error.</td>
</tr>
<tr>
<td>80</td>
<td>A stream binary piece with the same BLOB sequence number or stream identifier already exists.</td>
</tr>
</tbody>
</table>

The `proc_WriteStreamToRBS` stored procedure MUST NOT return a result set.
3.1.5.133 proc_WriteStreams

The proc_WriteStreams stored procedure creates a new stream binary piece for a document.

```
PROCEDURE dbo.proc_WriteStreams (    @SiteId       uniqueidentifier,    @DocId        uniqueidentifier,    @Partition    tinyint,    @StreamData   tvpStreamData READONLY,    @BSNData      tvpBSNMetadata2 READONLY,    @AddStream    bit,    @OverwriteStream bit )
```

@SiteId: The site collection identifier of the site collection containing the document.

@DocId: The document identifier of the document.

@Partition: The stream partition that the new stream binary piece belongs to.

@StreamData: Contains subsets of data for the stream binary piece, corresponding to the Data, Offset, and Length columns.

@BSNData: Contains the metadata for the subsets of data for the stream binary piece, corresponding to @StreamData. The subset data in a position of @StreamData MUST correspond to the BLOB sequence number and StreamId specified in the same position in @BSNData.

@AddStream: If this parameter is one and a stream binary piece already exists with the same BLOB sequence number or stream identifier specified by @BSNData, BLOB sequence number and StreamId, this stored procedure MUST return 80.

@OverwriteStream: If this parameter is one, @AddStream MUST be ignored.

Return Values: The proc_WriteStreams stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>2</td>
<td>The document associated with the stream binary pieces to add does not exist.</td>
</tr>
<tr>
<td>29</td>
<td>Unexpected error.</td>
</tr>
<tr>
<td>80</td>
<td>A stream binary piece with the BLOB sequence number or stream identifier specified from @BSNData already exists.</td>
</tr>
</tbody>
</table>

The proc_WriteStreams stored procedure MUST NOT return a result set.

3.1.5.134 proc_WriteStreamToSQL

The proc_WriteStreamToSQL stored procedure creates a new stream binary piece for a document.

```
PROCEDURE dbo.proc_WriteStreamToSQL (    @SiteId       uniqueidentifier,    @DocId        uniqueidentifier,    @Partition    tinyint,    @BSN         bigint,    @Size        int,    @StreamId    int )
```
@Type tinyint,
@Expiration datetime,
@Data0 varbinary(max), @Length0 bigint, @Offset0 bigint,
@Data1 varbinary(max), @Length1 bigint, @Offset1 bigint,
@Data2 varbinary(max), @Length2 bigint, @Offset2 bigint,
@Data3 varbinary(max), @Length3 bigint, @Offset3 bigint,
@Data4 varbinary(max), @Length4 bigint, @Offset4 bigint,
@Data5 varbinary(max), @Length5 bigint, @Offset5 bigint,
@Data6 varbinary(max), @Length6 bigint, @Offset6 bigint,
@Data7 varbinary(max), @Length7 bigint, @Offset7 bigint,
@Data8 varbinary(max), @Length8 bigint, @Offset8 bigint,
@Data9 varbinary(max), @Length9 bigint, @Offset9 bigint,
@Data10 varbinary(max), @Length10 bigint, @Offset10 bigint,
@Data11 varbinary(max), @Length11 bigint, @Offset11 bigint,
@Data12 varbinary(max), @Length12 bigint, @Offset12 bigint,
@Data13 varbinary(max), @Length13 bigint, @Offset13 bigint,
@Data14 varbinary(max), @Length14 bigint, @Offset14 bigint,
@Data15 varbinary(max), @Length15 bigint, @Offset15 bigint,
@Data16 varbinary(max), @Length16 bigint, @Offset16 bigint,
@Data17 varbinary(max), @Length17 bigint, @Offset17 bigint,
@Data18 varbinary(max), @Length18 bigint, @Offset18 bigint,
@Data19 varbinary(max), @Length19 bigint, @Offset19 bigint,
@Data20 varbinary(max), @Length20 bigint, @Offset20 bigint,
@Data21 varbinary(max), @Length21 bigint, @Offset21 bigint,
@Data22 varbinary(max), @Length22 bigint, @Offset22 bigint,
@Data23 varbinary(max), @Length23 bigint, @Offset23 bigint,
@Data24 varbinary(max), @Length24 bigint, @Offset24 bigint,
@Data25 varbinary(max), @Length25 bigint, @Offset25 bigint,
@Data26 varbinary(max), @Length26 bigint, @Offset26 bigint,
@Data27 varbinary(max), @Length27 bigint, @Offset27 bigint,
@Data28 varbinary(max), @Length28 bigint, @Offset28 bigint,
@Data29 varbinary(max), @Length29 bigint, @Offset29 bigint,
@Data30 varbinary(max), @Length30 bigint, @Offset30 bigint,
@Data31 varbinary(max), @Length31 bigint, @Offset31 bigint,
@Data32 varbinary(max), @Length32 bigint, @Offset32 bigint,
@Data33 varbinary(max), @Length33 bigint, @Offset33 bigint,
@Data34 varbinary(max), @Length34 bigint, @Offset34 bigint,
@Data35 varbinary(max), @Length35 bigint, @Offset35 bigint,
@Data36 varbinary(max), @Length36 bigint, @Offset36 bigint,
@Data37 varbinary(max), @Length37 bigint, @Offset37 bigint,
@Data38 varbinary(max), @Length38 bigint, @Offset38 bigint,
@Data39 varbinary(max), @Length39 bigint, @Offset39 bigint,
@Data40 varbinary(max), @Length40 bigint, @Offset40 bigint,
@Data41 varbinary(max), @Length41 bigint, @Offset41 bigint,
@Data42 varbinary(max), @Length42 bigint, @Offset42 bigint,
@Data43 varbinary(max), @Length43 bigint, @Offset43 bigint,
@Data44 varbinary(max), @Length44 bigint, @Offset44 bigint,
@Data45 varbinary(max), @Length45 bigint, @Offset45 bigint,
@Data46 varbinary(max), @Length46 bigint, @Offset46 bigint,
@Data47 varbinary(max), @Length47 bigint, @Offset47 bigint,
@Data48 varbinary(max), @Length48 bigint, @Offset48 bigint,
@Data49 varbinary(max), @Length49 bigint, @Offset49 bigint,
@Data50 varbinary(max), @Length50 bigint, @Offset50 bigint,
@Data51 varbinary(max), @Length51 bigint, @Offset51 bigint,
@Data52 varbinary(max), @Length52 bigint, @Offset52 bigint,
@Data53 varbinary(max), @Length53 bigint, @Offset53 bigint,
@Data54 varbinary(max), @Length54 bigint, @Offset54 bigint,
@Data55 varbinary(max), @Length55 bigint, @Offset55 bigint,
@SiteId: The site collection identifier of the site collection containing the document.

@DocId: The document identifier of the document.

@Partition: The stream partition that the new stream binary piece belongs to.

@BSN: The BLOB sequence number of the new stream binary piece.

@Size: The size, in bytes, of the new stream binary piece.

@Type: The stream type of the new stream binary piece.

@Expiration: The timestamp in UTC format of the stream binary piece. If the stream binary piece does not have an expiration, this parameter MUST be NULL.

@Data[0-63]: A piece of the binary data for the stream binary piece. See definition for details of format.

@Length[0-63]: This value MUST be the length of the corresponding @Data parameter, in bytes.

@Offset[0-63]: The offset in the stream binary piece where the corresponding @Data belongs. This value MUST be greater than or equal to zero, and this value plus the corresponding @Length parameter MUST be less than or equal to the total size of the new stream binary piece.

Return Values: The proc_WriteStreamToSQL stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>29</td>
<td>Unexpected error.</td>
</tr>
<tr>
<td>80</td>
<td>A stream binary piece with the same BLOB sequence number or stream identifier already exists.</td>
</tr>
</tbody>
</table>

The proc_WriteStreamToSQL stored procedure MUST NOT return a result set.

3.1.5.135 proc_SetStreamsToDoc

The proc_SetStreamsToDoc stored procedure associates stream binary pieces with a document.

PROCEDURE dbo.proc_SetStreamsToDoc (  
    @SiteId uniqueidentifier,  
    @DocId uniqueidentifier,  
    @HistVersion int,  
    @Level tinyint,  
    @Partition tinyint,  
    @StreamsToAdd tvpBSNMetadata READONLY,  
    @StreamsToDel tvpBSNMetadata READONLY,  
    @DeleteAllStreams bit,  
    @QuotaChange bigint = NULL OUTPUT  
)

@SiteId: The site collection identifier of the site collection containing the document.

@DocId: The document identifier of the document.

@HistVersion: If the document is a historical version, this parameter MUST be the UI version number. Otherwise, this parameter MUST be zero.
@Level: The publishing level of the document.

@Partition: The stream partition to which the stream binary pieces belong.

@StreamsToAdd: Specifies the BLOB sequence number and stream identifier for the stream binary pieces to associate with the document.

@StreamsToDelete: Specifies the BLOB sequence number for the stream binary pieces to unassociate with the document.

@DeleteAllStreams: If this parameter is one, all existing stream binary pieces MUST remove their association with the document.

@QuotaChange: An optional output parameter that specifies the quota change (negative value or zero), in bytes, of all the stream binary pieces that were unassociated from the document based on the @StreamsToDelete and @DeleteAllStreams parameters.

**Return Values:** The proc_SetStreamsToDoc stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>29</td>
<td>Unexpected error.</td>
</tr>
<tr>
<td>80</td>
<td>A stream binary piece with the same BLOB sequence number or stream identifier is already associated with the document.</td>
</tr>
</tbody>
</table>

The proc_SetStreamsToDoc stored procedure MUST NOT return a result set.

### 3.1.5.136 proc_SetStreamsToDocNoTVP

The proc_SetStreamsToDocNoTVP stored procedure associates stream binary pieces with a document.

```sql
PROCEDURE dbo.proc_SetStreamsToDocNoTVP (
    @SiteId uniqueidentifier,
    @DocId uniqueidentifier,
    @HistVersion int,
    @Level tinyint,
    @Partition tinyint,
    @BSNToAdd0 bigint, @StreamIdToAdd0 bigint,
    @BSNToAdd1 bigint, @StreamIdToAdd1 bigint,
    @BSNToAdd2 bigint, @StreamIdToAdd2 bigint,
    @BSNToAdd3 bigint, @StreamIdToAdd3 bigint,
    @BSNToAdd4 bigint, @StreamIdToAdd4 bigint,
    @BSNToAdd5 bigint, @StreamIdToAdd5 bigint,
    @BSNToAdd6 bigint, @StreamIdToAdd6 bigint,
    @BSNToAdd7 bigint, @StreamIdToAdd7 bigint,
    @BSNToAdd8 bigint, @StreamIdToAdd8 bigint,
    @BSNToAdd9 bigint, @StreamIdToAdd9 bigint,
    @BSNToAdd10 bigint, @StreamIdToAdd10 bigint,
    @BSNToAdd11 bigint, @StreamIdToAdd11 bigint,
    @BSNToAdd12 bigint, @StreamIdToAdd12 bigint,
    @BSNToAdd13 bigint, @StreamIdToAdd13 bigint,
    @BSNToAdd14 bigint, @StreamIdToAdd14 bigint,
    @BSNToAdd15 bigint, @StreamIdToAdd15 bigint,
    @BSNToAdd16 bigint, @StreamIdToAdd16 bigint,
    @BSNToAdd17 bigint, @StreamIdToAdd17 bigint,
    @BSNToAdd18 bigint, @StreamIdToAdd18 bigint,
    @BSNToAdd19 bigint, @StreamIdToAdd19 bigint,
    @BSNToAdd20 bigint, @StreamIdToAdd20 bigint,
)
@BSNToDelete0 bigint,
@BSNToDelete1 bigint,
@BSNToDelete2 bigint,
@BSNToDelete3 bigint,
@BSNToDelete4 bigint,
@BSNToDelete5 bigint,
@BSNToDelete6 bigint,
@BSNToDelete7 bigint,
@BSNToDelete8 bigint,
@BSNToDelete9 bigint,
@BSNToDelete10 bigint,
@BSNToDelete11 bigint,
@BSNToDelete12 bigint,
@BSNToDelete13 bigint,
@BSNToDelete14 bigint,
@BSNToDelete15 bigint,
@BSNToDelete16 bigint,
@BSNToDelete17 bigint,
@BSNToDelete18 bigint,
@BSNToDelete19 bigint,
@BSNToDelete20 bigint,
@BSNToDelete21 bigint,
@BSNToDelete22 bigint,
@BSNToDelete23 bigint,
@BSNToDelete24 bigint,
@BSNToDelete25 bigint,
@BSNToDelete26 bigint,
@BSNToDelete27 bigint,
@BSNToDelete28 bigint,
@BSNToDelete29 bigint,
@BSNToDelete30 bigint,
@BSNToDelete31 bigint,
@DeleteAllStreams bit,
@QuotaChange bigint = NULL OUTPUT
)

@SiteId: The site collection identifier of the site collection containing the document.

@DocId: The document identifier of the document.

@HistVersion: If the document is a historical version, this parameter MUST be the UI version number. Otherwise, this parameter MUST be zero.

@Level: The publishing level of the document.

@Partition: The stream partition to which the stream binary pieces belong.

@BSNToAdd[0-31]: The BLOB sequence number of the stream binary pieces to associate with the document.

@StreamIdToAdd[0-31]: The stream identifiers of the stream binary pieces to associate with the document. The position of the stream identifier in @StreamIdToAdd MUST match the position of the BLOB sequence number in @BSNToAdd.
@BSNToDelete[0-31]: The BLOB sequence number of the stream binary pieces to unassociate with the document.

@DeleteAllStreams: If this parameter is one, all existing stream binary pieces MUST remove their association with the document.

@QuotaChange: An optional output parameter that specifies the sum, in bytes, of all the stream binary pieces that were unassociated from the document based on the @BSNToDelete[0-31] and @DeleteAllStreams parameters.

Return Values: The proc_SetStreamsToDocNoTVP stored procedure returns an integer return code which MUST be listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
<tr>
<td>29</td>
<td>Unexpected error.</td>
</tr>
<tr>
<td>80</td>
<td>A stream binary piece with the same BLOB sequence number or stream identifier is already associated with the document.</td>
</tr>
</tbody>
</table>

The proc_SetStreamsToDocNoTVP stored procedure MUST NOT return a result set.

### 3.1.5.137 TVF_Docs_Url_Level

The TVF_Docs_Url_Level table value function is invoked to return a subset of rows from the Docs view (section 2.2.6.3).

```sql
FUNCTION [dbo].[TVF_Docs_Url_Level] (
    @SiteId uniqueidentifier,
    @DirName nvarchar(256),
    @LeafName nvarchar(128),
    @Level tinyint
)
```

@SiteId: All the rows returned from the Docs view MUST have SiteId = @SiteId.

@DirName: All the rows returned from the Docs view MUST have DirName = @DirName.

@LeafName: All the rows returned from the Docs view MUST have LeafName = @LeafName.

@Level: All the rows returned from the Docs view MUST have Level = @Level.

Return Values: TVF_Docs_Url_Level returns all rows from the Docs view that are specified as follows.

(SiteId = @SiteId AND DeleteTransactionId = 0x AND DirName = @DirName AND LeafName = @LeafName AND Level = @Level.)

### 3.1.5.138 TVF_UserData_ListItemLevelRow

The TVF_UserData_ListItemLevelRow table value function is invoked to return a subset of rows from the UserData View (section 2.2.6.8).

```sql
FUNCTION [dbo].[TVF_UserData_ListItemLevelRow] (  )
```
(  
  @tp_SiteId uniqueidentifier,
  @tp_ListId uniqueidentifier,
  @tp_Id int,
  @tp_Level tinyint,
  @tp_ROWOrdinal tinyint
)

@tp_SiteId: All the rows returned from the UserData View (section 2.2.6.8) MUST have tp_SiteId = @tp_SiteId.

@tp_ListId: All the rows returned from the UserData View (section 2.2.6.8) MUST have tp_ListId = @tp_ListId.

@tp_Id: All the rows return from the UserData View (section 2.2.6.8) MUST have tp_Id = @tp_Id.

@tp_Level: All the rows return from the UserData View (section 2.2.6.8) MUST have tp_Level = @tp_Level.

@tp_ROWOrdinal: All the rows returned from the UserData View (section 2.2.6.8) MUST have tp_ROWOrdinal = @tp_ROWOrdinal.

Return Values: TVF_UserData_ListItemLevelRow (section 3.1.5.137) returns all rows from the UserData View (section 2.2.6.8) that are specified as follows.

(tp_SiteId = @tp_SiteId AND tp_ListId = @tp_ListId AND tp_DeleteTransactionId = 0x AND
 tp_IsCurrentVersion = CONVERT(bit, 1) AND tp_Id = @tp_Id AND tp_CaculatedVersion = 0 AND
 tp_Level = @tp_Level AND tp_ROWOrdinal = @tp_ROWOrdinal)

3.1.5.139 TVF_UserData_PId_DId_Level_Row

The TVF_UserData_PId_DId_Level_Row table value function is invoked to return a subset of rows from the UserData View (section 2.2.6.8).

FUNCTION [dbo].[TVF_UserData_PId_DId_Level_Row]
(
    @tp_SiteId uniqueidentifier,
    @tp_ParentId uniqueidentifier,
    @tp_DocId uniqueidentifier,
    @tp_Level tinyint,
    @tp_ROWOrdinal tinyint
)

@tp_SiteId: All the rows returned from the UserData View MUST have tp_SiteId = @tp_SiteId.

@tp_ParentId: All the rows returned from the UserData View MUST have tp_ParentId = @tp_ParentId.

@tp_DocId: All the rows returned from the UserData View MUST have tp_DocId = @tp_DocId.

@tp_Level: All the rows returned from the UserData View MUST have tp_Level = @tp_Level.

@tp_ROWOrdinal: All the rows returned from the UserData View MUST have tp_ROWOrdinal = @tp_ROWOrdinal.

Return Values: TVF_UserData_PId_DId_Level_Row returns all rows from the UserData View that are specified as follows.
3.1.5.140  proc_HasCurrentPublishVersion

The proc_HasCurrentPublishVersion stored procedure is invoked to test whether the specified list item has a current published version.

PROCEDURE proc_HasCurrentPublishVersion {
    @SiteId     uniqueidentifier,
    @ListId     uniqueidentifier,
    @ItemId     int,
    @RequestGuid uniqueidentifier = null OUTPUT
};

@SiteId: The site collection identifier of the site collection.

@ListId: The list identifier of the list that contains the specified list item.

@ItemId: The list item identifier of the specified list item.

@RequestGuid: The optional request identifier for the current request.

Return values: This stored procedure returns an integer return code, which MUST be included in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The specified list item does not have a current published version.</td>
</tr>
<tr>
<td>1</td>
<td>The specified list item has a current published version.</td>
</tr>
</tbody>
</table>

This stored procedure MUST NOT return a result set.

3.1.5.141  proc_GetSiteDenyPermMask

The proc_GetSiteDenyPermMask stored procedure is invoked to return a WSS Rights Mask (section 2.2.2.15) specifying the rights to deny to the current user in the specified site collection.

PROCEDURE proc_GetSiteDenyPermMask {
    @SiteId     uniqueidentifier,
    @RequestGuid uniqueidentifier = NULL OUTPUT
};

@SiteId: The Site Collection Identifier (section 2.2.1.1.9) of the site collection for which to retrieve the deny rights mask.

@RequestGuid: The optional request identifier for the current request.

Return Values: This stored procedure MUST return an integer return code of 0.

This stored procedure MUST return the following result set.

3.1.5.141.1  Deny Permissions Mask Result Set
The **Deny Permissions Mask Result Set** returns a **WSS Rights Mask** (section 2.2.2.15) specifying the rights to deny to the current user in the specified site collection.

{DenyPermMask}: A **WSS Rights Mask** (section 2.2.2.15) specifying the rights to deny to the current user.

### 3.1.5.142 proc_GetNewListItemId

The **proc_GetNewListItemId** stored procedure is called to get the next available row identifier of a list (1) and the number of rows that will be copied or moved when copying or moving a URL to that list (1). The T-SQL syntax for the stored procedure is as follows:

```sql
PROCEDURE proc_GetNewListItemId(
    @SiteId                     uniqueidentifier,
    @FullUrl                    nvarchar(260),
    @NewListId                  uniqueidentifier,
    @bIsCopy                    bit,
    @NewDocLibRowId             int OUTPUT,
    @MaxNewRows                 int OUTPUT
);
```

@SiteId: The site collection identifier of the **site collection** which contains the URL specified by @FullUrl.

@FullUrl: The full URL to copy or move.

@NewListId: The list identifier of the target list.

@bIsCopy: A bit flag that indicates whether to copy or move the URL specified by @FullUrl. The value MUST be 1 to copy or MUST be zero to move.

@NewDocLibRowId: This is an output parameter whose value indicates the next available row identifier in the target list (1) where the URL specified by @FullUrl is being copied or moved to. The return value MUST be obtained by calling **proc_GenerateNextId** when the URL specified by @FullUrl is a file or folder; otherwise is undefined.

@MaxNewRows: This is an output parameter whose value indicates the total number of items as specified by @FullUrl that will be copied or moved. The return value MUST be 1 if the URL specified by @FullUrl is a file; MUST be the number of items contained in the folder if the URL specified by @FullUrl is a folder; otherwise is undefined.

Return values: An integer that MUST be zero.

### 3.1.5.143 proc_UpdateDiskUsed

The **proc_UpdateDiskUsed** stored procedure is called to update the disk-used size, in Bytes, of a site collection.

The T-SQL syntax for the stored procedure is as follows:

```sql
PROCEDURE proc_UpdateDiskUsed{
    @SiteId                     uniqueidentifier,
    @bUpdateTimeStampForce     bit = 0
};
```
@SiteId: The site collection identifier of the site collection whose disk-used size is to update.

@bUpdateTimeStampForce: A bit flag specifying whether to update the time stamp of LastContentChange of the site collection. If the value is 1, LastContentChange of the site collection MUST be updated with the current time. If the value is 0, LastContentChange MUST be unmodified.

Return Code Values: This stored procedure MUST return 0 upon completion.

Result Sets: MUST NOT return any result sets.

3.1.6 Timer Events

If the execution timeout event (2) is triggered, then the execution of the stored procedure is terminated and the call fails.

3.1.7 Other Local Events

None.

3.2 Web Front End Client Details

The front-end Web server acts as a client when it calls the back-end database server requesting execution of stored procedures.

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The front-end Web server can maintain the following sets of data for this protocol within object structures. There is no requirement for the data within these structures to be a complete representation of all data maintained on the back-end database server, but can be populated as various requests to the back-end database server are fulfilled. Data maintained on the front-end Web server can be discarded after individual sequences of requests have completed as part of a response for a higher level event.

- Configuration
- Site Collections
- Sites
- Lists
- List Items
- Documents
- Users
- Groups
3.2.2 Timers

A connection timeout timer is set up on the front-end Web server to govern the total connection time for any requests to the back-end database server. The amount of time is governed by a timeout value configured on the front-end Web server for all back-end database server connections.

3.2.3 Initialization

The front-end Web server MUST validate the user making the request before calling the stored procedure(s). The Site Collection Identifier and the User Identifier for the user making the request are looked up by the front-end Web server before calling additional stored procedure(s).

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The front-end Web server handles each stored procedure with the same processing method of calling the stored procedure and waiting for the result code and any result sets that will be returned.

The front-end Web server can execute dynamically generated SQL queries against the stored procedures or the tables and views used within the database. However, unless otherwise specified, any data addition, removal, or modification MUST occur only by calling the listed stored procedure. SQL queries MUST NOT attempt to add, remove, or update data in any table or view in the content databases or configuration database (section 3.1.1), unless explicitly described in this section.

3.2.6 Timer Events

If the connection timeout event (2) is triggered, the connection and the stored procedure call fails.

3.2.7 Other Local Events

None.
4 Protocol Examples

This section provides specific example scenarios for end-to-end file and permissions management against WSS. These examples describe in detail the process of communication between the various server components involved in the WSS deployment. In conjunction with the detailed protocol documentation described in the reference documents, this information is intended to provide a comprehensive view of how WSS front-end Web server communicates with both EUC and BEDS systems.

4.1 File: Open File OM

This example describes the requests and responses made when the front-end Web server opens an existing file stored as a document in a document library on the BEDS with a SharePoint Object Model call.

Figure 2: File: Open File OM

This scenario is initiated by a call to the Microsoft.SharePoint.SPFile.OpenBinary() object model command. For simplicity's sake, this example assumes that the file is stored as a document in a document library, and that the requested version is a draft created by the same user who is opening the file. This example assumes that:

- The code has already instantiated the site collection (SPSite), Site (SPWeb), and document library (SPList) objects containing the document to be opened.
- Auditing is disabled for the site collection.
- The current user has File Open permissions for the document.
- Site groups in the site collection do not include any domain groups as members.

The following actions happen:

1. The WFE builds a dynamic query that invokes the proc_FetchDocForHttpGet stored procedure.
2. The BEDS returns a return code of 0, and returns the following result sets:
HTTP Document Metadata Result Set (section 3.1.5.19.1). This returns the document metadata needed to further process the document.

Domain Group Cache Versions Result Set (section 2.2.4.4). The versions of the Domain Group cache on BEDS and WFE, used to determine if either the BEDS or WFE has more up-to-date information about external group membership in Roles, which are stored as site groups.

Domain Group Cache WFE Update Result Set (section 2.2.4.5). Used to update the WFE's external group map cache if needed. Under our assumptions, this result set is empty, because no domain groups are members of any roles in the site collection.

User Information Result Set (section 3.1.5.19.5). Used to establish that the current user has permissions to open the file.

Document Content Stream Result Set (section 3.1.5.19.9). Includes the document stream containing the binary file content for the current version of the document visible to the user, along with additional document metadata.

Site Audit Mask Result Set (section 2.2.4.22). Under our assumptions, auditing is not enabled on the site collection, so the SiteGlobalAuditMask column is NULL.

List Audit Mask Result Set (section 3.1.5.19.11), containing auditing information for the document's containing document library. Under our assumptions, auditing is not enabled for the site collection, so the fields containing audit masks are NULL.

A Dynamic Query Result Set containing a single row with a single unnamed column, holding the value of the @Level output parameter from the stored procedure proc_FetchDocForHttpGet. The value is 2, indicating that the content returned is from the latest draft version.

3. The OM returns control to the calling program with the array of bytes for the document stream of the requested file.

4.1.1 Determine a User's Permission Level to a Document

This example describes the requests made to determine a user's permissions on a document in a document library.

This example assumes:

- The user is authenticated and not anonymous.
- The user is a member of a role defined on the site (2).
- The document exists in a document library.

This scenario is initiated by a call to the object model command SPLListItem.EffectiveBasePermissions().

The following actions happen:

1. The WFE builds a dynamic query that invokes the proc_FetchDocForHttpGet stored procedure.

2. The BEDS returns a return code of 0, and returns a number of result sets, including the HTTP Document Metadata Result Set (section 3.1.5.19.1). The HTTP Document Metadata Result Set contains an ACL which specifies the permissions on this document.

3. The OM returns control to the calling program with the SPBasePermissions object containing the permissions the current user has on the requested document.
4.2 Group Add User to Site Group OM

This example describes the requests made when a user is added to a site group using SharePoint Object Model code running on the WFE.

Figure 3: Group Add User To Site Group OM

This scenario is initiated by a call to the object model command `SPGroup.Users.Add()`. For simplicity's sake, this example assumes that:

- The code has already instantiated the site collection (SPSite), Web (SPWeb), and Group (SPGroup) objects, which contain the site group for this session.
- The instantiated objects have not yet populated information about the user information list.
- The user to be added to the group is currently a user in the site collection.

The following actions happen:

1. The WFE first fetches the properties for the "User Information List" of the target site collection, a SharePoint list that contains information about users and groups (2) registered in a site
collection. It does this by calling the stored procedure `proc_GetListMetaDataAndEventReceivers` (section 3.1.5.33) using TDS.

2. The BEDS returns two result sets, which include the List Metadata (section 2.2.4.14) and Event Receivers (section 2.2.4.11) for the specified user information list.

3. The WFE determines the site map for the site collection by calling the configuration database stored procedure `proc_getSiteMapBySiteId` (section 3.1.5.39) using TDS.

4. The BEDS returns a single Site Map By Id Result Set (section 3.1.5.39.1), which includes the site map for the specified site collection.

5. If the WFE determines that the user information list has not been populated in the current SPSite object, it requests the list field information for the site collection's user information list by calling the `proc_GetListFields` (section 3.1.5.32) stored procedure using TDS.

6. The BEDS returns a single Fields Information Result Set (section 3.1.5.32.1), which includes the field information for the specified user information list.

7. If the WFE determines that it needs to populate the list of SPFeatures, the list of child objects that inherit from the base class SPFeatureDefinition will be populated by calling the configuration database stored procedure `proc_getObjectsByBaseClass` (section 3.1.5.35) using TDS.

8. The BEDS returns a single Object ID Result Set (section 2.2.4.18), which includes a list of child object identifiers for the specified base class and parent object.

9. The WFE builds a transactional dynamic SQL query to add the user to the site collection and add the user's property information to the user information list. This query is sent to the SQL server using TDS. On the SQL server the following actions occur:

   - The query begins a new SQL transaction.
   - The query attempts to add the user to the site collection using the stored procedure `proc_SecAddUser` (section 3.1.5.51).
   - The query checks if the user exists in the site collection's user information list.
   - If the user is not found in the site collection's user information list, the query attempts to add the user's properties to the site collection's user information list using the stored procedure `proc_AddListItem` (section 3.1.5.4).
   - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

10. The BEDS returns a single result set indicating the status of the actions within the query and the output parameters from the `proc_SecAddUser` command.

11. The WFE queries the list of users in the user information list for the site collection by building a SQL Batch call to the UserData View (section 2.2.6.8), which is then sent to the SQL server using TDS.

12. The BEDS returns a single result set with a list of items in the UserData View.

13. The WFE builds a transactional dynamic SQL query to add or update the list item that represents the user with the user's properties. This query is sent to the SQL server using TDS. On the SQL server the following actions occur:

   - The query begins a new SQL transaction.
   - The query checks if the user exists in the user information list for the site collection.
- If the user is not found in the user information list for the site collection, then the query attempts to add the user's properties to the site collection's user information list using the stored procedure `proc_AddListItem`.

- Otherwise, if the user is found in the site collection's user information list, the query attempts to update the user's properties in the site collection's user information list using the stored procedure `proc_UpdateListItem` (section 3.1.5.127).

- The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

14. The BEDS returns two result sets, which contain the output and return codes from the Add List Item or Update List Item commands.

15. The WFE adds the user to the specified site group by calling the stored procedure `proc_SecAddUserToSiteGroup` (section 3.1.5.52) using TDS.

16. The BEDS supplies a return code indicating success or failure of the procedure.

### 4.3 Group Update Site Group Properties OM

This example describes the interactions made when properties are updated for a particular site group.
Figure 4: Group Update Site Group Properties OM

This scenario is initiated by a call to the object model command `SPGroup.Update()`. For simplicity’s sake, this example assumes that:

- the code has already instantiated the site collection (`SPSite`) and site (`SPWeb`) objects for this session.
- the site group to be updated is in the site collection and is a member of the site (2).

The following actions happen:

1. The WFE determines if the site (2) has the directory management service enabled with a call to the Configuration Database stored procedure `proc_GetObjectsByClass` (section 3.1.5.36) using TDS.

2. The BEDS returns a single Object ID Result Set (section 2.2.4.18) row, which includes a value set if the directory management service is enabled.

3. The WFE fetches the properties for the target site collection’s user information list, a SharePoint list containing information about users and groups (2) registered in a site collection. It does this by calling the stored procedure `proc_GetListMetaDataAndEventReceivers` using TDS.
4. The BEDS returns two result sets, which include the List Metadata (section 2.2.4.14) and Event Receivers (section 2.2.4.11) for the specified user information list.

5. If the WFE determines that the user information list has not been populated in the current SPSite object, it requests the list field information for the site collection's user information list by calling the proc_GetListFields stored procedure using TDS.

6. The BEDS returns a single Field Information Result Set, which includes the field information for the specified user information list.

7. If the WFE determines that it needs to populate the list of SPFeatures, the list of child objects that inherit from the base class SPFeatureDefinition is populated by calling the configuration database stored procedure proc_getObjectsByBaseClass using TDS.

8. The BEDS returns a single Object ID Result Set, which includes a list of child object identifiers for the specified base class and parent object.

9. The WFE builds a dynamic SQL query to select existing information for the site group using TDS.

10. The BEDS returns a single result set, which includes existing data for the user.

11. The WFE builds a transactional dynamic SQL query to update the site group information in the site collection. This query is sent to the SQL server using TDS. On the SQL server the following actions occur:

   - The query begins a new SQL transaction.
   - The query attempts to load the site group information using the stored procedure proc_SecSetSiteGroupProperties (section 3.1.5.115).
   - If the site group is not in the list of site members, the query invokes proc_AddListItem with the updated information.
   - Otherwise, the query attempts to update the site group’s properties in the site collection user information list, using the stored procedure proc_UpdateListItem (section 3.1.5.127).
   - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

12. The BEDS returns a single result set, which indicates the return code status of the actions within the query.

**4.4 Security: Add User to Document Library via Object Model**

This example describes the requests made when a user is added to the "contributor" role of a document library that has its own scope for independently managed permissions.
This scenario is initiated by a call to the object model command `SPRoleAssignmentCollection.Add(SPRoleAssignment)`.

For simplicity's sake, this example assumes that the code has already instantiated the necessary site collection (SPSite), site (SPWeb), and list (SPList) objects, as well as the role (SPRoleDefinition), role bindings (SPRoleDefinitionBindingCollection) and role assignment (SPRoleAssignment) objects, to construct a representation of the role within the document library to which the user will be added.

1. The WFE first fetches the properties for the user information list of the site collection. The user information list is a SharePoint list containing information about users in the site collection. It does this by calling the stored procedure `proc_GetListMetaDataAndEventReceivers` (section 3.1.5.33) using TDS.

2. The BEDS returns two result sets. The **List Metadata Result Set** returns a single row of data with the metadata for the list. The second result set is the **Event Receivers Result Set** (section 2.2.4.11). In this example, there are no registered event receivers, and so zero rows are returned.

3. The WFE builds a transactional dynamic SQL query to add the user to the document library's "Contributor" role and to add or update the user's property information in the user information list. This query is sent to the SQL server using TDS. On the SQL server, the following actions occur:

   - The query begins a new SQL transaction.
   - The query attempts to add the user to the site collection's user information list using the stored procedure `proc_SecAddUser` (section 3.1.5.51).
   - The query checks if the user exists in the site collection's user information list.
   - If the user is not found in the site collection's user information list, then the query attempts to add the user's properties to the site collection's user information list using the stored procedure `proc_AddListItem`.
   - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

4. The BEDS returns a single result set indicating the status of the actions within the query and the output parameters from the `proc_SecAddUser` stored procedure.
5. The WFE fetches the list's scope ACL and anonymous user permission information by calling the stored procedure proc_SecGetAclFromScope (section 3.1.5.62) using TDS.

6. The BEDS returns an ACL and Permissions Result Set (section 2.2.4.1), which lists ACL and permission information for the list's scope.

7. The WFE adds the user to the list's "contributor" role assignment membership site group by calling the stored procedure proc_SecAddPrincipalToRole (section 3.1.5.49) using TDS.

8. The BEDS returns a Site Audit Mask Result Set (section 2.2.4.22), which lists Auditing Flags information for the site collection and document library.

4.5 Security: Break Web Inheritance OM

This example describes the requests made to create unique security role assignments for a site (2), rather than inheriting them from a parent.

![Figure 6: Break Web Inheritance OM](image)

This scenario is initiated by a call to the object model command `spweb.BreakRoleInheritance(true)`. For simplicity's sake, this example assumes that:

- the code has already instantiated the site collection (SPSite), the parent site and child subsite (SPWeb) objects for this session, and
- the child subsite is initially in the same scope as its parent site (2).

The following actions happen:

1. The WFE first retrieves metadata for the requested child site (2). It does this by calling the proc_GetTpWebMetaDataAndListMetaData (section 3.1.5.40) stored procedure using TDS.

2. The BEDS returns the following five result sets:
   - Web URL Result Set (section 3.1.5.40.1). This contains the store-relative form URL of the root of the requested child site (2).
- **Domain Group Cache Versions Result Set** (section 2.2.4.4). This contains information about the version numbers associated with the External Group Map Cache for the requested site (2).

- **Domain Group Cache WFE Update Result Set** (section 2.2.4.5). This returns the binary data needed to refresh the external group Map Cache.

- **Site Metadata Result Set** (section 2.2.4.24). This contains metadata for the requested site (2).

- **Event Receivers Result Set** (section 2.2.4.11). This contains information about the event receivers defined for the requested site (2).

3. The WFE then retrieves security permissions information about the requested site (2). It does this by calling the **proc_SecGetSecurityInfo** (section 3.1.5.83) stored procedure using TDS.

4. The BEDS returns the **Security Information Result Set** (section 3.1.5.83.1), which consists of information about security permissions about the requested site (2).

5. The WFE then builds a Dynamic SQL Query to convert the requested site (2) to use unique permissions (as opposed to inheriting those permission from the Parent site). It does this by calling the **proc_SecChangeToUniqueScope** (section 3.1.5.56) stored procedure using TDS.

6. The BEDS returns the following two result sets:
   - **SiteAudit Mask Result Set** (section 2.2.4.22), containing information about the Audit Flags (section 2.2.2.1) set for the requested child site (2).
   - A Dynamic SQL Result Set, containing the Scope Identifier (section 2.2.1.1.8) of the new scope generated for the child site (2).

### 4.6 Site Collection Lookup

To allow SharePoint's data storage to scale out, site collections can be stored in many content databases. This example illustrates the protocol operations between the client (front-end Web server) and server (back-end database server) needed to find and connect to a specific content database given a site collection URL. An existing connection to the configuration database using lower-level protocols is assumed.

#### 4.6.1 Retrieving the Farm Id

The example begins by calling **proc_GetObjectsByClass** (section 3.1.5.36) with the Farm Class ID.

![Figure 7: Retrieving the Farm ID](image)

This call returns a result set including the **Configuration Object ID** of the Farm Configuration Object. **proc_getObjectsByClass** can return result sets with multiple rows, but this implementation of the
protocol only ever stores one configuration object (section 2.2.5.1) with the Farm Class ID and ignores all but the first row if multiple rows are returned.

4.6.2 Retrieving the Alternate URL Collection Ids

Next, the Farm Configuration Object Id and the Alternate URL Collection Class Id are passed to proc_getObjectsByBaseClass.

Figure 8: Web Service Lookup Operations

Because the implementation defined the returned Farm Configuration Object to be the Parent of all Alternate URL Collections, this call returns the Configuration Object Ids of all Alternate URL Collections stored in the Configuration Database.

4.6.3 Retrieving the Alternate URL Collections

The retrieved Alternate URL Collection Configuration Object Ids are then passed to proc_getObject to retrieve the full contents of the Alternate URL Collection Configuration Objects.

Figure 9: Retrieving the Alternate URL Collections

4.6.4 Alternate URL Matching

At this point, the client can execute the XPath query against the properties of each of the Alternate URL Collections (section 4.6.3) returned to extract all of the alternate URLs. Each of these URLs is then compared against the portion of the incoming URL beginning with the scheme component and ending with the authority component (for example, “http://example.com:80”). If a match is found the configuration object ID of the Alternate URL Collection containing the matching URL is stored for later use as the request Alternate URL Collections. Otherwise, the site collection lookup operation terminates.

4.6.5 Retrieving the Web Service Ids

Next, the Farm Configuration Object Id returned and the specified Web Service Class Id are passed to proc_getObjectsByBaseClass.
4.6.6 Retrieving the Web Application Ids

In this step, the specified Web application Class Id is passed to \texttt{proc_getObjectsByBaseClass} with each of the retrieved Web Service Configuration Object Ids. This returns the Configuration Object Ids of all Web applications in the Configuration Database.

4.6.7 Retrieving the Web Applications

The retrieved Web Application Configuration Object Ids are then passed to \texttt{proc_getObject} to retrieve the full contents of the Web Application Configuration Objects.

4.6.8 Web Application Lookup

At this point, the client can execute the Web Application Alternate URL Collection XPath Query to extract the Alternate URL Collection IDs (section 2.2.5.1.7.1) associated with each retrieved Web...
application. These Alternate URL Collection IDs are compared against the Request Alternate URL Collection ID. If a match is found, the associated Web application is used as the Request Web application for the remainder of site collection lookup. Otherwise, site collection lookup terminates.

4.6.9 Prefix Matching

Web applications contains a set of site collection Prefixes which contain a name and a type. Site collection lookup extracts these values from the Request Web Application Properties using XPath Queries.

The Prefix names are URL Path Components used to determine which portion of the incoming URL Path Component is the server-relative URL of the site collection. This is done by matching all of the Prefixes in the Request Web application against the start of the Path Component of the incoming URL. If more than one Prefix matches the beginning of the incoming URL Path Component, the longest matching Prefix is used.

There are two types of Prefix: wildcard and explicit. A Web application can contain any combination of both types.

4.6.9.1 Explicit Prefixes

An explicit prefix indicates that the portion of the Path Component up to and including the Prefix are included in the site collection server-relative URL. For example, if a user requests http://example.com/sitename/web/list/document.htm and if the Web application corresponding to http://example.com contains an explicit prefix named "sitename", then "/sitename" is the server-relative URL of the site collection.

<table>
<thead>
<tr>
<th>Incoming URL</th>
<th>Web Application Explicit Prefixes</th>
<th>Resulting Site Collection Server-Relative URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://example.com/a/b/c.htm">http://example.com/a/b/c.htm</a></td>
<td>&quot;a&quot;</td>
<td>&quot;/a&quot;</td>
</tr>
<tr>
<td><a href="http://example.com/a/b/c.htm">http://example.com/a/b/c.htm</a></td>
<td>&quot;a&quot;, &quot;a/b&quot;</td>
<td>&quot;/a/b&quot;</td>
</tr>
<tr>
<td><a href="http://example.com/a/b.htm">http://example.com/a/b.htm</a></td>
<td>&quot;a&quot;, &quot;a/b&quot;</td>
<td>&quot;/a&quot;</td>
</tr>
<tr>
<td><a href="http://example.com/a/b.htm">http://example.com/a/b.htm</a></td>
<td>&quot;c&quot;</td>
<td>&lt;No Match&gt;</td>
</tr>
<tr>
<td><a href="http://example.com/a/b.htm">http://example.com/a/b.htm</a></td>
<td>&quot;&quot;</td>
<td>&quot;/&quot;</td>
</tr>
</tbody>
</table>

4.6.9.2 Wildcard Prefixes

A wildcard prefix indicates that the portion of the Path Component up to and including the first Path Component segment following the Prefix are included in the site collection name. For example, if a user makes a request for http://example.com/sites/sitename/web/list/document.htm, and if the Web application corresponding to http://example.com contains a Wildcard Prefix named "sites", then "/sites/sitename" is the server-relative URL of the site collection.

<table>
<thead>
<tr>
<th>Incoming URL</th>
<th>Web Application Wildcard Prefixes</th>
<th>Resulting Site Collection Server-Relative URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://example.com/a/b/c/d.htm">http://example.com/a/b/c/d.htm</a></td>
<td>&quot;a&quot;, &quot;a/b&quot;</td>
<td>&quot;/a/b/c&quot;</td>
</tr>
<tr>
<td><a href="http://example.com/a/b.htm">http://example.com/a/b.htm</a></td>
<td>&quot;a&quot;, &quot;a/b&quot;</td>
<td>&lt;No Match&gt;</td>
</tr>
<tr>
<td><a href="http://example.com/a/b.htm">http://example.com/a/b.htm</a></td>
<td>&quot;&quot;</td>
<td>&quot;/a&quot;</td>
</tr>
</tbody>
</table>
4.6.10 Site Collection Id Lookup

Once the site collection URL is determined, it is passed to proc_getSiteMap (section 3.1.5.38), along with the Request Web Application ID.

A site collection ID is returned along with the configuration object ID of the content database in which the site collection content is stored. If the specified combination site collection URL and Web application ID cannot be found in the configuration database, the site collection does not exist and site collection lookup terminates.

4.6.11 Building Content Database Connection String

At this point, the only step that remains is to establish a connection to the content database. This requires a content database connection string, which is built by combining the following components.

4.6.11.1 Name

Once the content database ID is known, it is passed to proc_GetObject, which returns the content database configuration object (section 2.2.5.1).

The name field of the content database configuration object is used in the connection string as the content database name.

Credentials

The optional Username and Password are extracted from the Properties of the Content Database configuration object using XPath queries.

Instance
The ParentId of the content database configuration object is passed in another call to procGetObject, which returns the database service instance configuration object, the name field of which is used as the Database Server Instance.

![Database Server Instance](image1)

**Figure 15: Database Server Instance**

Server Address

The ParentId of the Database Service Instance Configuration Object is then passed to the one final call to procGetObject, which a server configuration object. The name field of the server configuration object is used as the Server Address component of the connection string.

![Server Address](image2)

**Figure 16: Server Address**

At the end of this step, the incoming URL has been successfully translated into a site collection URL, a site collection Id, and a Content Database Connection String. These components are then ready to be used to call other stored procedures in this and other protocols.

### 4.7 User Update User Properties OM

This example describes the interactions made when site member properties are updated for a particular user.
Figure 17: User Update User Properties OM

This scenario is initiated by a call to the object model command `SPUser.Update()` . For simplicity’s sake, this example assumes that:

- the code has already instantiated the site collection (`SPSite`) and Web (`SPWeb`) Objects for this session, and
- the user to be updated is in the site collection and a member of the site (2).

The following actions happen:

1. The WFE fetches the properties for the target site collection's user information list, a SharePoint list containing information about users and groups (2) registered in a site collection. It does this by calling the stored procedure `proc_GetListMetaDataAndEventReceivers` (section 3.1.5.33) using TDS.

2. The BEDS returns two result sets, which include the list metadata (section 2.2.4.14) and event receivers for the specified user information list.

3. If the WFE determines that the user information list has not been populated in the current `SPSite` object, it requests the list field information for the site collection's user information list by calling the stored procedure `proc_GetListFields` using TDS.
4. The BEDS returns a single fields information result set, which includes the field information for the specified user information list.

5. If the WFE determines that it needs to populate the list of SPFeatures, the list of child objects that inherit from the base class SPFeatureDefinition is populated by calling the configuration database stored procedure proc_getObjectsByBaseClass using TDS.

6. The BEDS returns a single Object ID result set, which includes a list of child object identifiers for the specified base class and parent object.

7. The WFE builds a dynamic SQL query to select existing information for the user using TDS.

8. The BEDS returns a single result set, which includes existing data for the user.

9. The WFE builds a transactional dynamic SQL query to update the user information in the site (2). This query is sent to the SQL server using TDS. On the SQL server the following actions occur:
   - The query begins a new SQL transaction.
   - The query attempts to update the user information using the stored procedure proc_SecUpdateUser (section 3.1.5.121).
   - The query attempts to update the user’s properties in the site’s user information list using the stored procedure proc_UpdateListItem (section 3.1.5.127).
   - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

10. The BEDS returns a single result set, which indicates the return code status of the actions within the query.

11. The WFE builds a dynamic SQL query to select updated information for the user using TDS.

12. The BEDS returns a single result set, which includes the data for the user.

13. The WFE builds a transactional dynamic SQL query to update the user information in the site (2). This query is sent to the SQL server using TDS. On the SQL server the following actions occur:
   - The query begins a new SQL transaction.
   - The query attempts to update the site collection’s user list data using the stored procedure proc_UpdateListItem.
   - Then the query attempts to update the user’s properties in the site collection’s user information table using the stored procedure proc_UpdateUserInfoInTableFromRowUpdater (section 3.1.5.129).
   - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

14. The BEDS returns a successful return code status.

4.8 Version Negotiation

The following scenario is an example of the protocol version negotiation sequence between a WFE and a content database where the content database version does not match that expected by the WFE.
1. The WFE calls the **proc_GetVersion** (section 3.1.5.42) stored procedure in the content database on the BEDS, with the @VersionId parameter set to '6333368D-85F0-4EF5-8241-5252B12B2E50'.

2. The **proc_GetVersion** stored procedure returns the output parameter @Version with a value of '3.0.149.0'. The application has a pre-defined version of '0', which reflects the current state of the software. Any new database created by this application will have this version. The application determines that the version from the database is smaller (meaning it is older) than the version the WFE expects, and does not perform any further operations against the database.

### 4.9 Folder: Move Folder OM

This example describes the requests and responses made when a folder is moved within a list using an object model call.
• the current user has permissions to view the folder.

The following actions happen:

1. The front-end Web server builds a dynamic query that invokes the `proc_SecUpdateUserActiveStatus` stored procedure.

2. The front-end Web server builds a dynamic query that invokes the `proc_SecGetIndividualUrlSecurityCheckEventReceivers` stored procedure against the source folder.

3. The back-end database server returns a return code of 0, and returns the following result sets:
   - **Individual URL Security Result Set.** This returns security information about the specific source folder.

4. The front-end Web server builds a dynamic query that invokes the `proc_SecGetIndividualUrlSecurityCheckEventReceivers` stored procedure against the destination folder.

5. The back-end database server returns a return code of 0, and returns the following result sets:
   - **Individual URL Security Result Set.** This returns security information about the specific destination folder.

6. The front-end Web server builds a dynamic query that invokes the `proc_GetNewListItemId` stored procedure.

7. The front-end Web server builds a transactional dynamic SQL query to move the folder.
   - The query begins a new SQL transaction.
   - The query attempts to rename the URL for the folder using the `proc_RenameUrl` stored procedure. The back-end database server returns a return code of 0, and returns the **Rename Result Set**, which gives basic information about the old and new URLs for the renamed folder.
   - The query attempts to get **Audit Flags** information about the new folder using the `proc_GetAuditMaskOutput` stored procedure.
   - The query rolls back the SQL transaction if any of the previous procedures were not successful.
   - Assuming the previous procedures were successful, the query updates the disk spaced used for the **site (2)** by calling the `proc_UpdateDiskUsed` stored procedure, and commits the transaction
5 Security

5.1 Security Considerations for Implementers

Security for this protocol is controlled by the access rights to the databases on the back-end database server, which is negotiated as part of the Tabular Data Stream [MS-TDS] protocol.

In a trusted subsystem model, the process running on the web front end server uses its own security principal identity to access the content database on the back-end database server on behalf of the user, rather than using the account of the user accessing the web front end as a database access account to access the content database. The database access account used by the web front end server needs to have access to the content database on the back-end database server. If the account does not have the correct access rights, access will be denied when attempting to set up the [MS-TDS] connection to the content database, or when calling the stored procedures.

5.2 Index of Security Parameters

<table>
<thead>
<tr>
<th>Security Parameter</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc_SecAddPrincipalToRole</td>
<td>3.1.5.49</td>
</tr>
<tr>
<td>proc_SecAddRoleDef</td>
<td>3.1.5.50</td>
</tr>
<tr>
<td>proc_SecAddUser</td>
<td>3.1.5.51</td>
</tr>
<tr>
<td>proc_SecAddUserToSiteGroup</td>
<td>3.1.5.52</td>
</tr>
<tr>
<td>proc_SecAddWebMembership</td>
<td>3.1.5.53</td>
</tr>
<tr>
<td>proc_SecChangeToInheritedList</td>
<td>3.1.5.54</td>
</tr>
<tr>
<td>proc_SecChangeToInheritedWeb</td>
<td>3.1.5.55</td>
</tr>
<tr>
<td>proc_SecChangeToUniqueScope</td>
<td>3.1.5.56</td>
</tr>
<tr>
<td>proc_SecCheckDeletedAccounts</td>
<td>3.1.5.57</td>
</tr>
<tr>
<td>proc_SecCloneRoleDefinitions</td>
<td>3.1.5.58</td>
</tr>
<tr>
<td>proc_SecCreateSiteGroup</td>
<td>3.1.5.59</td>
</tr>
<tr>
<td>proc_SecDecCurrentUsersCount</td>
<td>3.1.5.60</td>
</tr>
<tr>
<td>proc_SecGetAccountStatus</td>
<td>3.1.5.61</td>
</tr>
<tr>
<td>proc_SecGetAclFromScope</td>
<td>3.1.5.62</td>
</tr>
<tr>
<td>proc_SecGetAllAclsForSite</td>
<td>3.1.5.63</td>
</tr>
<tr>
<td>proc_SecGetAllGroupsAndMembershipInfo</td>
<td>3.1.5.64</td>
</tr>
<tr>
<td>Proc_SecGetApplicationPrincipalAndUserToken</td>
<td>3.1.5.65</td>
</tr>
<tr>
<td>proc_SecGetCompleteWebRoleMemberList</td>
<td>3.1.5.66</td>
</tr>
<tr>
<td>proc_SecGetCurrentUsersCount</td>
<td>3.1.5.67</td>
</tr>
<tr>
<td>proc_SecGetDomainGroupMapData</td>
<td>3.1.5.68</td>
</tr>
<tr>
<td>proc_SecGetGroupById</td>
<td>3.1.5.69</td>
</tr>
<tr>
<td>Security Parameter</td>
<td>Section</td>
</tr>
<tr>
<td>--------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>proc_SecGetGroupOwner</td>
<td>3.1.5.70</td>
</tr>
<tr>
<td>proc_SecGetGroupSecurityScopes</td>
<td>3.1.5.71</td>
</tr>
<tr>
<td>proc_SecGetIndividualUrlSecurityCheckEventReceivers</td>
<td>3.1.5.72</td>
</tr>
<tr>
<td>Proc_SecGetItemsWithUniquePermissions</td>
<td>3.1.5.73</td>
</tr>
<tr>
<td>proc_SecGetPrincipalByEmail</td>
<td>3.1.5.74</td>
</tr>
<tr>
<td>proc_SecGetPrincipalById</td>
<td>3.1.5.75</td>
</tr>
<tr>
<td>Proc_SecGetPrincipalByIdEx</td>
<td>3.1.5.76</td>
</tr>
<tr>
<td>proc_SecGetPrincipalByLogin</td>
<td>3.1.5.77</td>
</tr>
<tr>
<td>proc_SecGetPrincipalByLogin20</td>
<td>3.1.5.78</td>
</tr>
<tr>
<td>proc_SecGetPrincipalDisplayInformation20</td>
<td>3.1.5.79</td>
</tr>
<tr>
<td>Proc_SecGetRoleAssignments</td>
<td>3.1.5.80</td>
</tr>
<tr>
<td>proc_SecGetRoleBindingsForAllPrincipals</td>
<td>3.1.5.81</td>
</tr>
<tr>
<td>proc_SecGetRoleDefs</td>
<td>3.1.5.82</td>
</tr>
<tr>
<td>proc_SecGetSecurityInfo</td>
<td>3.1.5.83</td>
</tr>
<tr>
<td>proc_SecGetSiteAdmins</td>
<td>3.1.5.84</td>
</tr>
<tr>
<td>proc_SecGetSiteGroupById</td>
<td>3.1.5.85</td>
</tr>
<tr>
<td>proc_SecGetSiteGroupByTitle</td>
<td>3.1.5.86</td>
</tr>
<tr>
<td>proc_SecGetSiteGroupByTitle20</td>
<td>3.1.5.87</td>
</tr>
<tr>
<td>proc_SecGetUserAccountDirectoryPath</td>
<td>3.1.5.88</td>
</tr>
<tr>
<td>proc_SecGetUserPermissionOnGroup</td>
<td>3.1.5.89</td>
</tr>
<tr>
<td>Proc_SecGetWebsAndListsWithUniquePermissions</td>
<td>3.1.5.90</td>
</tr>
<tr>
<td>proc_SecListAllSiteMembers</td>
<td>3.1.5.91</td>
</tr>
<tr>
<td>proc_SecListAllWebMembers</td>
<td>3.1.5.92</td>
</tr>
<tr>
<td>proc_SecListGroupsInRole</td>
<td>3.1.5.93</td>
</tr>
<tr>
<td>proc_SecListScopeGroups</td>
<td>3.1.5.94</td>
</tr>
<tr>
<td>proc_SecListScopeUsers</td>
<td>3.1.5.95</td>
</tr>
<tr>
<td>proc_SecListSiteGroupMembership</td>
<td>3.1.5.96</td>
</tr>
<tr>
<td>proc_SecListSiteGroups</td>
<td>3.1.5.97</td>
</tr>
<tr>
<td>proc_SecListSiteGroupsContainingUser</td>
<td>3.1.5.98</td>
</tr>
<tr>
<td>proc_SecListSiteGroupsWhichUserOwns</td>
<td>3.1.5.99</td>
</tr>
<tr>
<td>proc_SecListUsersInRole</td>
<td>3.1.5.100</td>
</tr>
<tr>
<td>proc_SecMigrateUser</td>
<td>3.1.5.101</td>
</tr>
<tr>
<td>Security Parameter</td>
<td>Section</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>proc_SecRecalculateWebFGP</td>
<td>3.1.5.102</td>
</tr>
<tr>
<td>proc_SecRefreshToken</td>
<td>3.1.5.103</td>
</tr>
<tr>
<td>proc_SecRemoveGroup</td>
<td>3.1.5.104</td>
</tr>
<tr>
<td>proc_SecRemovePrincipalFromScope</td>
<td>3.1.5.106</td>
</tr>
<tr>
<td>proc_SecRemoveRoleDef</td>
<td>3.1.5.107</td>
</tr>
<tr>
<td>proc_SecRemoveUserFromScopeByLogin</td>
<td>3.1.5.108</td>
</tr>
<tr>
<td>proc_SecRemoveUserFromSite</td>
<td>3.1.5.109</td>
</tr>
<tr>
<td>proc_SecRemoveUserFromSiteGroup</td>
<td>3.1.5.110</td>
</tr>
<tr>
<td>proc_SecRemoveUserFromSiteGroupByLogin</td>
<td>3.1.5.111</td>
</tr>
<tr>
<td>proc_SecResetItemPerm</td>
<td>3.1.5.112</td>
</tr>
<tr>
<td>proc_SecResetWebToDefaultRoleDefinition</td>
<td>3.1.5.113</td>
</tr>
<tr>
<td>proc_SecResolvePrincipal</td>
<td>3.1.5.114</td>
</tr>
<tr>
<td>proc_SecSetSiteGroupProperties</td>
<td>3.1.5.115</td>
</tr>
<tr>
<td>proc_SecSetUserAccountDirectoryPath</td>
<td>3.1.5.116</td>
</tr>
<tr>
<td>proc_SecSetWebRequestAccess</td>
<td>3.1.5.117</td>
</tr>
<tr>
<td>proc_SecUpdateAnonymousPermMask</td>
<td>3.1.5.118</td>
</tr>
<tr>
<td>proc_SecUpdateDomainGroupMapData</td>
<td>3.1.5.119</td>
</tr>
<tr>
<td>proc_SecUpdateRoleDef</td>
<td>3.1.5.120</td>
</tr>
<tr>
<td>proc_SecUpdateUser</td>
<td>3.1.5.121</td>
</tr>
<tr>
<td>proc_SecUpdateUserActiveStatus</td>
<td>3.1.5.122</td>
</tr>
</tbody>
</table>
6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

- Microsoft SharePoint Foundation 2013
- Microsoft SharePoint Server 2016

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 2.2.1.1.10: For example, by default, SharePoint Foundation 2013 creates three implementation-specific site groups with default site group identifiers for use in role assignments when provisioning a new site collection: "Site Owners" (3), "Site Visitors" (4), and "Site Members" (5).

<2> Section 2.2.2.8: This flag is not used in SharePoint Foundation 2013.

<3> Section 2.2.7.3.3.2: SharePoint Foundation 2013 allows a filtering user interface for fields with the aggregation attribute set to merge, but only in list views where Filter=1 occurs in the view's query parameter.

<4> Section 2.2.7.3.3.2: SharePoint Foundation 2013 does include this attribute with a value of 0.

<5> Section 2.2.7.3.3.2: SharePoint Foundation 2013 sets the value MinusSign for this field.

<6> Section 3.1.5.42: SharePoint products use text-based SQL queries to get the version number which are equivalent in functionality to proc_GetVersion.
7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Revision class</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.3.8 WSS Compressed Structures</td>
<td>Updated description for Version and FileHeaderSize fields.</td>
<td>Major</td>
</tr>
<tr>
<td>3.1.5 Message Processing Events and Sequencing Rules</td>
<td>Added the three stored procedures in the table.</td>
<td>Major</td>
</tr>
</tbody>
</table>
8 Index

A

Abstract data model
  client 451
  server 212
ACL and Permission result set 69
App Principal Information result set 101
App Principal Permissions result set 102
Applicability 29
Attribute groups - overview 210
Attributes - overview 210

B

Binary structures
  Calendar View Options Type 63
  External Group Token 64
  Token Group Offset and Attributes 65
  Token Groups 65
  WSS ACE 66
  WSS ACL Format 66
  WSS Compressed Structures 67
  WSS External Group Map Cache Format 67
  WSS External Group Record 68
  WSS User Token 68

Calendar View Options Type binary structure 63
Capability negotiation 30
Change tracking 476
CHOICEDEFINITION - complex type 192
CHOICEDEFINITIONS - complex type 192
Client
  abstract data model 451
  higher-layer triggered events 452
    initialization 452
    local events 452
    message processing 452
    sequencing rules 452
    timer events 452
    timers 452
Common data types
  overview 31
Complex types
  CHOICEDEFINITION 192
  CHOICEDEFINITIONS 192
  FieldDefinition 193
  FieldDefinitionDatabase 205
  FieldDefinitionDatabaseWithVersion 205
  FieldDefinitionTP 206
  FieldParserRef 206
  FieldParserRefs 207
  FieldRefDefinitionField 207
  FieldRefDefinitionIndex 208
  FieldRefDefinitionTP 208
  IndexDefinitionTP 209
  MAPPINGDEFINITION 209
  MAPPINGDEFINITIONS 209
  Custom Actions From Scope result set 69

D

Data model - abstract
  client 451
  server 212
Data types
  common 31
    Time Zone Identifier simple type 45
Data types - simple
  Time Zone Identifier 45
  Document Content Metadata result set 71
  Document Content Stream result set 73
  Document Metadata result set 73
  Document Version Metadata result set 77
  Domain Group Cache BEDS Update result set 70
  Domain Group Cache Versions result set 70
  Domain Group Cache WFE Update result set 71

E

Elements - overview 210
Empty result set 81
Event Receivers result set 81
Events
  local - client 452
  local - server 451
  timer - client 452
  timer - server 451
Examples
  File - determine a user's permission level to a document 454
  File - open file OM 453
  Folder - move folder OM 470
  Group add user to site group OM 455
  Group update site group properties OM 457
  overview 453
  Security - add user to document library via object model 459
  Security - break web inheritance OM 461
  Site collection lookup 462
  Site collection lookup - alternate URL matching 463
  Site collection lookup - building content database connection string 466
  Site collection lookup - prefix matching 465
  Site collection lookup - retrieving the alternate URL collection identifiers 463
  Site collection lookup - retrieving the alternate URL collections 463
  Site collection lookup - retrieving the farm identifier 462
  Site collection lookup - retrieving the web application identifiers 464
  Site collection lookup - retrieving the web applications 464
  Site collection lookup - retrieving the web service identifiers 463
  Site collection lookup - site collection identifier lookup 466
  Site collection lookup - web application lookup 464
  User update user properties OM 467

477 / 484
Version negotiation 469
External Group Token binary structure 64

F

FALSE_Case_Insensitive_Else_Anything - simple type 185
FieldAggregationAttribute - simple type 187
FieldDefinition - complex type 193
FieldDefinitionDatabase - complex type 205
FieldDefinitionDatabaseWithVersion - complex type 205
FieldDefinitionTP - complex type 206
FieldInternalType - simple type 185
FieldParserRef - complex type 206
FieldParserRefs - complex type 207
FieldRefDefinitionField - complex type 207
FieldRefDefinitionIndex - complex type 208
FieldRefDefinitionTP - complex type 208
FieldRefType - simple type 188
FieldRichTextMode - simple type 189
Fields - vendor-extendible 30
File - determine a user’s permission level to a document example 454
File - open file OM example 453
File operations overview (synopsis) 28
fn_GetFullUrl method 222
Folder - move folder OM example 470

G

Glossary 15
Group add user to site group OM example 455
Group update site group properties OM example 457
Groups - overview 210

H

Higher-layer triggered events
client 452
server 213

I

IMEMode - simple type 189
Implementer - security considerations 472
Index of security parameters 472
IndexDefinitionTP - complex type 209
Individual URL Security result set 83
Informative references 28
Initialization
client 452
server 213
IntPositive - simple type 189
Introduction 15

J

JoinType - simple type 190

L

Link Information result set 85

List Metadata result set 85
List Related Fields result set 101
List Web Parts result set 90
Local events
client 452
server 451

M

MAPPINGDEFINITION - complex type 209
MAPPINGDEFINITIONS - complex type 209
Messages
ACL and Permission result set 69
App Principal Information result set 101
App Principal Permissions result set 102
attribute groups 210
attributes 210
Calendar View Options Type binary structure 63
CHOICEDEFINITION complex type 192
CHOICEDEFINITIONS complex type 192
common data types 31
Custom Actions From Scope result set 69
Document Content Metadata result set 71
Document Content Stream result set 73
Document Metadata result set 73
Document Version Metadata result set 77
Domain Group Cache BEDS Update result set 70
Domain Group Cache Versions result set 70
Domain Group Cache WFE Update result set 71
elements 210
Empty result set 81
Event Receivers result set 81
External Group Token binary structure 64
FALSE_Case_Insensitive_Else_Anything simple type 185
FieldAggregationAttribute simple type 187
FieldDefinition complex type 193
FieldDefinitionDatabase complex type 205
FieldDefinitionDatabaseWithVersion complex type 205
FieldDefinitionTP complex type 206
FieldInternalType simple type 185
FieldParserRef complex type 206
FieldParserRefs complex type 207
FieldRefDefinitionField complex type 207
FieldRefDefinitionIndex complex type 208
FieldRefDefinitionTP complex type 208
FieldRefType simple type 188
FieldRichTextMode simple type 189
groups 210
IMEMode simple type 189
IndexDefinitionTP complex type 209
Individual URL Security result set 83
IntPositive simple type 189
JoinType simple type 190
Link Information result set 85
List Metadata result set 85
List Related Fields result set 101
List Web Parts result set 90
MAPPINGDEFINITION complex type 209
MAPPINGDEFINITIONS complex type 209
namespaces 185
Overview (synopsis) 28

null

Namespaces 185
Normative references 27
NULL Individual URL Security result set 91
NULL Unique Permissions result set 92

null

Object ID result set 92
proc_SecAddWebMembership method 348
proc_SecAddWebToInheritedList method 348
proc_SecAddWebToInheritedWeb method 349
proc_SecChangeToUniqueInherited method 350
proc_SecCheckDeletedAccounts method 352
proc_SecCloneRoleDefinitions method 353
proc_SecCreateSiteGroup method 354
proc_SecCurrentUsersCount method 355
proc_SecGetAccountStatus method 356
proc_SecGetAccountScope method 356
proc_SecGetAllAclsForSite method 357
proc_SecGetAllGroupsAndMembershipInfo method 358
proc_SecGetApplicationPrincipalAndUserToken method 359
proc_SecGetCompleteWebRoleMemberList method 360
proc_SecGetCurrentUsersCount method 361
proc_SecGetDomainMapData method 362
proc_SecGetGroupById method 363
proc_SecGetGroupOwner method 364
proc_SecGetGroupSecurityScopes method 364
proc_SecGetIndividualUrlSecurityCheckEventRec method 365
proc_SecGetItemsWithUniquePermissions method 367
proc_SecGetPrincipalByEmail method 368
proc_SecGetPrincipalByEmail method 369
proc_SecGetPrincipalByIdEx method 370
proc_SecGetPrincipalByLogin method 372
proc_SecGetPrincipalByLogin20 method 373
proc_SecGetPrincipalDisplayInformation20 method 374
proc_SecGetRoleAssignments method 377
proc_SecGetRoleBindingsForAllPrincipals method 377
proc_SecGetCurrentRoleDef method 378
proc_SecGetSecurityInfo method 379
proc_SecGetSiteAdmins method 380
proc_SecGetSiteGroupByid method 381
proc_SecGetSiteGroupByTitle method 382
proc_SecGetSiteGroupByTitle20 method 382
proc_SecGetUserAccountDirectoryPath method 383
proc_SecGetUserPermissionOnGroup method 384
proc_SecGetWebsAndListWithUniquePermissions method 385
proc_SecListAllSiteMembers method 387
proc_SecListAllWebMembers method 388
proc_SecListGroupsInRole method 389
proc_SecListScopeGroups method 389
proc_SecListScopeUsers method 390
proc_SecListSiteGroupMembership method 390
proc_SecListSiteGroups method 391
proc_SecListSiteGroupsContainingUser method 392
proc_SecListSiteGroupsWhichUserOwns method 392
proc_SecListUsersInRole method 393
proc_SecMarkGroupForWebDelete method 397
proc_SecMigrateUser method 394
proc_SecRecalculateWebFgp method 395
proc_SecRefreshToken method 395
proc_SecRemoveGroup method 396
proc_SecRemovePrincipalFromScope method 399
proc_SecRemoveRoleDef method 400
proc_SecRemoveUserFromScopeByLogin method 400
proc_SecRemoveUserFromSite method 402
proc_SecRemoveUserFromSiteGroup method 402

proc_SecRemoveUserFromSiteGroupByLogin method 403
proc_SecResetItemPerm method 405
proc_SecResetWebToDefaultRoleDefinition method 405
proc_SecResolvePrincipal method 407
proc_SecSetSiteGroupProperties method 409
proc_SecSetUserAccountDirectoryPath method 411
proc_SecSetWebRequestAccess method 411
proc_SecUpdateAnonymousPermMask method 412
proc_SecUpdateDomainGroupMapData method 412
proc_SecUpdateRoleDef method 413
proc_SecUpdateUser method 414
proc_SecUpdateUserActiveStatus method 415
proc_SetsItemPerm method 444
proc_SetsWebToDoc method 444
proc_TakeOverCheckOut method 416
proc_UncheckoutDocument method 416
proc_UpdateDiskUsed method 450
proc_UpdateDocBuildDependencySet method 418
proc_UpdateDocument method 419
proc_UpdateListItem method 425
proc_UpdateListSettings method 430
proc_UpdateUserInTableFromRowUpdater method 434
proc_UriToWebUrl method 436
proc_WriteChunkToAllDocStreams method 436
proc_WriteStreams method 442
proc_WriteStreamToRBS method 441
proc_WriteStreamToSQL method 442
Product behavior 475

R

References 27
informative 28
normative 27

Relationship to other protocols 29
RelationshipDeleteBehaviorAttribute - simple type 191

Result sets - messages

ACL and Permission 69
App Principal Information 101
App Principal Permissions 102
Custom Actions From Scope 69
Document Content Metadata 71
Document Content Stream 73
Document Metadata 73
Document Version Metadata 77
Domain Group Cache BEDS Update 70
Domain Group Cache Versions 70
Domain Group Cache WFE Update 71
Empty 81
Event Receivers 81
Individual URL Security 83
Link Information 85
List Metadata 85
List Related Fields 101
List Web Parts 90
NULL Individual URL Security 91
NULL Unique Permissions 92
Object ID 92
Principal User Information 92
Server Time 93
Single Doc Link Information 93

481 / 484

[MS-WSSFO3] - v20170620
Windows SharePoint Services (WSS): File Operations Database Communications Version 3 Protocol
Copyright © 2017 Microsoft Corporation
Release: June 20, 2017
Proc_SecListAllWebMembers method 388
Proc_SecListGroupsInRole method 389
Proc_SecListScopeGroups method 389
Proc_SecListScopeUsers method 390
Proc_SecListSiteGroupMembership method 390
Proc_SecListSiteGroups method 391
Proc_SecListSiteGroupsContainingUser method 392
Proc_SecListSiteGroupsWhichUserOwns method 392
Proc_SecMarkGroupForWebDelete method 397
Proc_SecMigrateUser method 394
Proc_SecReCalculateWebFGP method 395
Proc_SecRefreshToken method 395
Proc_SecRemoveGroup method 396
Proc_SecRemovePrincipalFromScope method 399
Proc_SecRemoveRoleDef method 400
Proc_SecRemoveUserFromScopeByLogin method 400
Proc_SecRemoveUserFromSite method 402
Proc_SecRemoveUserFromSiteGroup method 402
Proc_SecRemoveUserFromSiteGroupByLogin method 403
Proc_SecResetItemPerm method 405
Proc_SecResetWebToDefaultRoleDefinition method 405
Proc_SecResolvePrincipal method 407
Proc_SecSetSiteGroupProperties method 409
Proc_SecSetUserAccountDirectoryPath method 411
Proc_SecSetWebRequestAccess method 411
Proc_SecUpdateAnonymousPermMask method 412
Proc_SecUpdateDomainGroupMapData method 412
Proc_SecUpdateRoleDef method 413
Proc_SecUpdateUser method 414
Proc_SecUpdateUserActiveStatus method 415
Proc_SetStreamsToDoc method 444
Proc_SetStreamsToDocNoTVP method 445
Proc_TakeOverCheckOut method 416
Proc_UncheckoutDocument method 416
Proc_UpdateDiskUsed method 450
Proc_UpdateDocBuildDependencySet method 418
Proc_UpdateDocument method 419
Proc_UpdateListItem method 425
Proc_UpdateListSettings method 430
Proc_UpdateUserInfoInTableFromRowUpdater method 434
Proc_UntoWebUrl method 436
Proc_WriteChunkToAllDocStreams method 436
Proc_WriteStreams method 442
Proc_WriteStreamToRBS method 441
Proc_WriteStreamToSQL method 442
Sequencing rules 213
timer events 451
timers 212
tvf Docs_Url _Level method 447
tvf UserData_ListItemLevelRow method 447
tvf UserData_Pid_Old _Level_Row method 448
Server Time result set 93
Simple data types
Time Zone Identifier 45
Simple types
FALSE CaseInsensitiveElseAnything 185
FieldAggregationAttribute 187
FieldInternalType 185
FieldRefType 188
FieldRichTextMode 189
IMEMode 189
IntPositive 189
JoinType 190
RelationshipDeleteBehaviorAttribute 191
TextDirection 190
TRUE CaseSensitiveElseAnything 191
TRUE If Present 190
TRUETYPE 191
UniqueIdentifierWithBracesUppercase 191
UniqueIdentifierWithOrWithoutBraces 191
Single Doc Link Information result set 93
Site Audit Mask result set 94
Site collection lookup - alternate URL matching example 463
Site collection lookup - building content database connection string example 466
Site collection lookup - prefix matching example 465
Site collection lookup - retrieving the alternate URL collection identifiers example 463
Site collection lookup - retrieving the alternate URL collections example 463
Site collection lookup - retrieving the farm identifier example 462
Site collection lookup - retrieving the web application identifiers example 464
Site collection lookup - retrieving the web applications example 464
Site collection lookup - retrieving the web service identifiers example 463
Site collection lookup - site collection identifier lookup example 466
Site collection lookup - web application lookup example 464
Site collection lookup example 462
Site Feature List result set 94
Site Metadata result set 95
Site MetaInfo result set 100
Standards assignments 30
Structures
XML 185
T
TextDirection - simple type 190
Time Zone Identifier simple type 45
timer events
client 452
server 451
timers
client 452
server 212
Token Group Offset and Attributes binary structure 65
Token Groups binary structure 65
Tracking changes 476
Transport 31
Triggered events - higher-layer
client 452
server 213
TRUE CaseSensitiveElseAnything - simple type 191
TRUE If Present - simple type 190